Design, Modelling and Control of a Continuum Manipulator Based on Multilayer Planar Springs

Qi, Peng

Awarding institution:
King's College London

KING'S College LONDON

DESIGN, MODELLING AND CONTROL OF A CONTINUUM MANIPULATOR BASED ON MULTILAYER PLANAR SPRINGS

Peng Qi

Department of Informatics

King's College London

This thesis is submitted for the degree of

Doctor of Philosophy

September 2015

To my loving parents

Qi Lidong & Jia Xiaoying

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text.

Peng Qi

September 2015, London

Abstract

There is a surge of research interest in the field of “continuum robotics”. Robots created under this paradigm offer many advantages and represent unique features in terms of flexibility, dexterity, safety and weight reduction. In the thesis, a novel continuum manipulator that integrates multiple layers of compliant planar springs is introduced – a structure that provides several notable advantages over existing designs. It possesses precise linear large-displacement motion and demonstrates effectively decoupling bending from contraction and thus reduces the uncontrolled compression when generating normal deflections; besides, an enlarged workspace of the end-effector is achieved by varying the length of the continuum manipulator via contraction.

The mechanics of the proposed continuum manipulator is investigated. An analytical method is provided to study the compliance characteristics of planar springs and derive the unified compliance matrix to represent the force-deflection relationships, allowing an accurate motion prediction. Differences of the compliance characteristics with respect to design variations of planar springs are discussed. An analysis regarding behaviours of the full continuum manipulator is given. According to the constant-curvature approximation, two kinematic models corresponding to three-tendon-driven and single-tendon-driven continuum manipulators are presented. This modelling methodology permits closed-form kinematics and also facilitates the derivation of differential kinematics and real-time control.

In view of the model’s complexity and uncertainty of the continuum manipulator, a fuzzy control approach is implemented for autonomous execution of end-effector motion tasks. The system state-space model is constructed using the general continuum manipulator kinematics with the constant-curvature assumption. The fuzzy controller is designed utilizing state-feedback control techniques. Thus, this control methodology enables a low-computation solution to this motion control problem without the need for continuously updating the Jacobian of the continuum manipulator. Besides, compared to traditional Jacobian-based controllers that suffer from model inaccuracies, the fuzzy control exhibits superior performances with respect to a specified cost function.

Acknowledgements

First and foremost I would like to express my greatest appreciation and gratitude to my supervisor Professor Kaspar Althoefer. He has invested an enormous amount of time and energy in me, and his invaluable insights and suggestions helped me at various stages of my research. He has been supportive and I remain indebted for his continuous encouragement. My sincere gratitude is also reserved for Dr. Hongbin Liu for his guidance and all the useful discussions. He has always been helpful and shared his experience and vast knowledge with me.

I would like to thank Professor Jian S. Dai for leading me to his passionate world of kinematics and mechanisms. I was fortunate and grateful to receive so much support and inculcation from him. His theory book becomes my bible. I am also very grateful to Professor Lakmal Seneviratne for getting me started in the field of continuum robotics. His scientific advice and knowledge and many insightful discussions and suggestions have guided my research. I own my gratitude to Dr. Hak-Keung Lam for providing so much valuable advice and immense knowledge on the motion control of continuum manipulators. I also learned from his modesty and rigorous scholarship. I will forever be thankful to Dr. Ali Ataollahi for teaching me many hands-on skills and inspiring me with the planar-spring-based designs. He is a real mentor at the early stage of my PhD study.

I have felt at home at KCL. I am grateful to all my friends and colleagues at the Centre of Robotics Research of King's College London. Especially, Xinsheng Zhang has helped and taught me immensely with his vase knowledge; Chuang Liu is the wisest fellow and most helpful friend I met. Chen Qiu, Jie Sun, Bo Xiao have always been friendly helping me with the theoretical analysis, hardware, and some mathematical concepts. I would also like to take the opportunity to thank Ahmad Ataka, Ali Shafti, Dr. Anuradha Ranasinghe, Shan Luo, Angela Faragasso, Dr. Guowu Wei, Dr. Jichun Li, Dr. Ketao Zhang, Dr. Sina Sareh, Xiaozhan Yang, Yan Yu, Yanbin Zhao, Yuandi Li, for the fun and supportive environment they created all these years. My thanks are also due to my cousin Minghui Jia, Yu Nie, Wenqian Cheng, and all the lovely people that I met in London, for bringing me a wonderful life in the beautiful city.

During the period of my PhD study, I have received the scholarships or grants from the following funding sources: the Chinese Scholarship Council, Department of Informatics (KCL), Graduate School (KCL), IEEE Robotics and Automation Society, the Great Britain-China Educational Trust on behalf of the Han Suyin Trust, and Henry Lester Trust Limited. They are also gratefully acknowledged.

Finally, I would like to express my deepest gratitude to my family for their love, encouragement, and the most solid support.

Table of Contents

Table of Figures

Table of Tables

Abbreviations

Abbreviation	Meaning
2D	Two-dimensional
3D	Three-dimensional
ABS	Acrylonitrile butadiene styrene
CAD	Computer-aided design
D-H	Denavit-Hartenberg
DOF	Degree of freedom
DPI	Dots per inch
FEA	Finite element analysis
FEM	Finite element modelling
F-S	Frenet-Serret
F/T	Force/Torque
IAE	Integral absolute error
LQR	Linear-quadratic regulator
MAE	Mean absolute error
MIS	Minimally invasive surgery
NiTi	Nickel-Titanium
PDC	Parallel distributed compensation
ROS	Robot operating system
se(3)	Lie algebra of *SE*(3)
*se**(3)	Dual Lie algebra of *SE*(3)
SE(3)	Special Euclidean group
so(3)	Lie algebra of *SO*(3)
SO(3)	Special orthogonal group
SOS	Sum of squares
SPD	Symmetric positive definite

Chapter 1 Introduction

1.1 Introduction

Continuum manipulators, inspired by elephant trunks and octopus tentacles, increasingly arouse the attention of researchers due to the appealing advantages such as compliance, dexterity and potential for miniaturized profile. A continuum manipulator has the capability to continuously bend and theoretically possesses an infinite number of degrees of freedom (DOFs). Remarkable developments in this area have been seen in the last two decades: various forms of design were created; many new applications for such continuum robots were demonstrated across different sectors, including industrial operations and health care environments (Walker, 2013). The related scientific problems range from designing and modelling continuum manipulators to low-level control and high-level task execution (Escande, Chettibi, Merzouki, Coelen, & Pathak, 2015).

Historically, the first continuum manipulator is generally accepted to be Anderson and Horn's tensor arm manipulator invented in the late 1960s (Anderson & Horn, 1967) – a tendon-driven spine-like flexible arm. Subsequently in 1971, Hirose started to propose creative designs of snake-like robots and appropriate control systems based on the biomechanical study of snakes (Hirose, 1993). The late 1990s and the 2000s saw an increasing trend of miniature continuum manipulators being moved into robotic surgery with a view to finding solutions for robot-assisted minimally invasive surgery (MIS) with its inherent access problems through small incisions (Camarillo, Krummel, & Salisbury, 2004), (Kanno, Haraguchi, Yamamoto, Tadano, & Kawashima, 2015).

Lots of efforts have been made on fabrication, material research, and design/construction of continuum manipulators. Super-elastic NiTi tubes/rods (Simaan, et al., 2009) and shape memory alloys (Ayvali & Desai, 2012) are the most common metals used to fabricate flexible arms. Less stiff plastics can also be seen to fabricate flexures or compliant segments using 3D printing technology. Other flexible members such as helical springs and bellows are utilized to construct skeleton arms and represent the desired flexibility characteristic as well. More sophisticated devices such as pneumatic artificial muscles and general pneumatic actuators provide mechanical softness via a flexible and inflatable membrane operated by gas pressure and have achieved numerous realizations in continuum manipulator designs. An example is the "Bionic Handling Assistant" – a project of Festo's Bionic Learning Network (Festo Corporate, 2010), which takes inspiration from the trunk of an elephant and is driven by a pneumatic system. Meanwhile, soft robotics as a subset of continuum robotics emerged with the development of novel soft actuators and sensors (Trivedi, Rahn, Kier, & Walker, 2008), (Girard, et al., 2015). With a soft

structure, these robots inherently enhanced the adaptability to interact with confined spaces or unstructured environments and perform delicate “whole arm manipulation” (Salisbury, 1987). Webster and Jones (Webster & Jones, 2010) presented a milestone review on the constant-curvature kinematics methodology of continuum manipulators and summarized the early developments. Most recently, Walker (Walker, 2013) reviewed the state of art of continuous backbone robot manipulators and analysed the hardware design principles.

Significant progress has not only been made in design but also in modelling, including both kinematics and dynamics. Early works include Chirikjian’s pilot research in the 1990s on establishing the fundamental modelling technique to formulate the dynamics of hyper-redundant manipulators (Chirikjian, 1994). Hannan and Walker provided the general kinematic model for continuum manipulators using the well-established Denavit-Hartenberg (D-H) convention (Hannan & Walker, 2003). This approach adopted the modelling methodology originally used for traditional rigid-link manipulators to establish the continuum kinematics via virtual rigid-link kinematics. Other approaches focusing on static modelling give insight into the mechanics of continuum manipulators based on the elastic beam theory (Rucker & Webster, 2011). Among different kinematic models, the underlying methodology is the use of constant-curvature approximation (Webster & Jones, 2010). It provides closed-form position and velocity kinematics which is the basis for real-time control and further motion planning.

There are several different approaches for the robotic control of continuum manipulators. Penning et al. investigated closed-loop control in both task space and joint space, resulting in improved end-point positioning accuracy of robot catheters (Penning, Jung, Borgstadt, Ferrier, & Zinn, 2011). Due to the nonlinear behaviour and high flexibility of continuum manipulators, the system performance has shown to benefit from closed-loop control. Regarding the selection of task space or joint space control, generally, the task space controllers that employ a feedback loop to directly minimize task errors show some advantages (Penning, Jung, Ferrier, & Zinn, 2012). In terms of kinematic control versus dynamic control, it is noted that kinematic control embedding the velocity-level kinematics is commonly utilised (Mahl, Hildebrandt, & Sawodny, 2014), (Bailly, Amirat, & Fried, 2011), (Chitrakaran, Behal, Dawson, & Walker, 2007); dynamic control has also been studied (Kapadia & Walker, 2011), however, the lack of a well understood efficient dynamic model of continuum manipulators limits its implementation. In (Penning & Zinn, 2014), considering the problems of steady state positioning errors and undesirable dynamic behaviours of continuum manipulators, a combined control system incorporating a position feedback and a modal-space controller was proposed and shown to be effective. At the intelligent control level, a distributed fuzzy controller was introduced as part of a control law in (Ivanescu, 2002) which

avoids the difficulties determined by the complexity of nonlinear integral-differential equations. Furthermore, a neural network-based tracking controller was presented for a wide range of continuum manipulators (Braganza, Dawson, Walker, & Nath, 2007) and it does not require an accurate manipulator dynamic model. In (Melingui, Lakhal, Daachi, Mbede, & Merzouki, 2015), an adaptive neural network controller was implemented achieving the end-effector position tracking control in real time with high accuracy. Likewise, considering the situation in which continuum manipulators interact with unknown obstacles and environments, a task space closed-loop controller, only based on empirical estimates of the real-time Jacobian but without using a model, is used for overcoming these disturbances (Yip & Camarillo, 2014). For the future, we foresee autonomous execution of command tracking tasks based on practical control strategies approaching more new applications in the presence of continuum manipulators. Recently, an impressive implementation of a motion controller for a catheter (realised as a type of a continuum manipulator) for beating heart intracardiac surgery has been reported in (Kesner & Howe, 2011), (Yuen, Kettler, Novotny, Plowes, & Howe, 2009).

1.2 Motivation

Continuum manipulators are a rapidly expanding class of robots. Design of robot structures is a fundamental issue and an active area of research in this field. The main design objective is to achieve inherent compliance for overall robot structure. Most examples focus on the bending properties and also it is desirable for the resulting backbone approximately complying with a constant curvature. However, most often this constant-curvature assumption is only validating for a certain range of bending angles due to the limited transversal material properties, for example, the elastic tube/rod-based structure (Simaan, 2005). On the other hand, in some situations, the bending effort is partially lost in compression. That is to say, when bending a continuum manipulator, the length of the continuum manipulator will also be inattentively changed. This coupling of bending and contraction makes the design difficulty to control, for examples, the helical-spring-based structure (Anderson & Horn, 1967) and the tendon-actuated pneumatic backbone (McMahan, Jones, & Walker, 2005). Upon the aforementioned facts, a type of planar spring (Howell L. L., 2001) is investigated, whose platform could perform both translational and rotational out-of-plane motions and meet the generalised Hooke's spring law. Subsequently, in order to construct the continuum manipulator with planar springs stacking in serial, the opposing connecting pattern of each adjacent planar spring layers is employed. This design concept remains both translational and rotational compliance properties without any structural restrictions, thus resulting in the continuum manipulator

with both 3D bending and contraction capabilities and cumulating the deformations of each layer. Due to the varying length of the proposed continuum structure, it achieves an enlarged workspace of tip motion, which prevents concatenating multiple sections.

The realisation of robotic motion control of continuum manipulators is important in enabling numerous real applications. However, due to the complexity of continuum manipulators' mathematical models and model inaccuracies, the progress of the effective control systems is limited. Traditional Jacobian-based controllers suffer from the high computational burden, because these methods require online updating of the Jacobian of the system or reply on continuously updated estimations of the Jacobian. Therefore, fuzzy control strategies incorporating linearized state-space models are explored to control the complex nonlinear system of the continuum manipulator. The objective is to lower the computational cost when executing real-time motion control and compensate for the modelling inaccuracies and the uncertainty inherently embodied in the physical structure of continuum manipulators.

1.3 Contributions

The central contributions of the research are summarised as follows:

1. A novel continuum manipulator design is presented in the context of continuum robotics. It consists of serially connected planar spring modules, each of which has the two-layer structure with three compliant legs. By the use of local planar springs in the design, a better (physical) performance form a continuum manipulator is obtained. This continuum manipulator has two degrees of freedom rotations and also possesses the ability to vary its length. It is the first time that both translational and rotational out-of-plane motions of the planar spring are utilised. Most importantly, the manipulator effectively decreases the contraction when generating bending deflections. This feature will enable easier controller design. Moreover, the essential design concept lies in that the bending of this continuum manipulator is not based on natural compliance of a continuous backbone element or soft skeletal elements but utilises the compliance of each structured planar spring module, thus, the manipulator demonstrates a cumulative large linear bending motion. With this inherent property underlying the physical structure, its kinematic model can be complied with the frequently used constant-curvature assumption which has the significant impact on modelling and real-time control of continuum robots. Furthermore, the manipulator can easily be manufactured using a 3D-printer. In the thesis,

a 3D-printed robot prototype made of multiple double-layer spine modules is used to conduct some experiments and illustrate the linearity and decoupled motions of the proposed manipulator, as well as the improved workspace compared to alternative spline designs. Besides, design variations of the planar spring by simply varying the angle of leg arrangement are studied, and planar springs with different configurations constitute a conic-shape continuum manipulator with the increasing flexibilities towards its distal end. This idea provides a potential to design the continuum manipulator with various bending configurations.

2. This thesis investigates the mechanics of the proposed continuum manipulator. The six-dimensional compliance characteristics of the utilised planar springs are studied using screw theory. The compliance matrix of the planar spring is developed with a bottom-up approach by treating the compliant layer as a type of hybrid flexure mechanism. This matrix reveals to be diagonal and the main compliance elements are validated in the FEA analysis. According to the developed diagonal compliance matrix, two important characteristics of the planar spring are further highlighted. One is the decoupled translational and rotational motion. This is because the entries outside the main diagonal are all zero, thus the applied force/moment only results in the intended translational/rotational motion, respectively, and no interactions. The other is isotropic rotational compliance, which means that a moment about any rotational axis passing through the centre of the planar spring layer will only produces the same angular displacement. With a comprehensive study of the compliance properties of the planar spring, analysis on the flexibility of the full robot is provided, which concludes that the bending deflection is decoupled with contraction. The kinematics of the continuum manipulator is established based on constant-curvature assumption. Two different cases are investigated, respectively corresponding to the three-tendon-driven continuum manipulator and the single-tendon-driven continuum manipulator. For both cases, the kinematic model and Jacobians are elucidated, forming the basis for control. The closed form dynamic models for our proposed type of continuum manipulators are established. All these theoretical studies with respect to the continuum manipulator are based on the mathematical framework and classical mechanics principles summarised in Chapter 2.
3. In view of the model complexities and the inevitable modelling inaccuracies, fuzzy control strategies are implemented for the continuum manipulator tip position control. This is the first time that fuzzy logic has been employed in this context. The basic fuzzy logic controller is implemented on the single-tendon-driven case, and the control objective is to regulate the tip point of a continuum manipulator to the target in 3D space. This task simulates the perceptual

docking of some frequently used medical continuum manipulators, for example, catheters. The performance of the proposed controller is validated in MATLAB simulations and proves the feasibility of the fuzzy control strategy for continuum manipulators. On the basis of the first trial of implementing the fuzzy controller, a fuzzy-model-based approach is then utilised to enable the continuum manipulator to accomplish autonomous execution of command tracking. Membership functions are employed to combine the linearized state-space models, to achieve, overall, a fuzzy model. The fuzzy model can aid the design of the fuzzy controller, especially when the controller design is guided by a stability analysis. In this case, the kinematic model of the continuum manipulator is assigned to be the general three-tendon-driven kinematics. This fuzzy-model-based control methodology is a hot research in recent years. The implementation of the fuzzy-mode-based tracking control for continuum manipulators in this thesis is the first that appears in the area of continuum robotics. The superior performances of this controller are validated in both MALTAB simulations and experiments compared to classical controllers found in the literature.

1.4 Contributed papers

The research presented in this thesis resulted in a number of publications listed as below:

Journal papers:

- P. Qi, C. Qiu, H. Liu, J. S. Dai, L. Seneviratne, and K. Althoefer, "A novel continuum manipulator design using serially connected double-layer planar springs," *IEEE/ASME Transactions on Mechatronics*, 2015 (DOI: 10.1109/TMECH.2015.2498738).
- P. Qi, C. Liu, A. Ataka, H. K. Lam, and K. Althoefer, "Kinematic control of continuum manipulators using a fuzzy-model-based approach," *IEEE Transactions on Industrial Electronics*, 2015 (Minor Revisions, IF = 6.5).
- C. Qiu, P. Qi, H. Liu, K. Althoefer, and J. S. Dai, "Analysis and validation of six-dimensional compliance of ortho-planar springs", *ASME Transactions: Journal of Mechanical Design*, 2015 (Accepted).
- P. Qi, H. Liu, L. Seneviratne, and K. Althoefer, "A conic shape-optimizable continuum robot arm using variations of multi-layer planar springs", *ASME Transactions: Journal of Mechanisms and Robotics*, 2015 (Under preparation, IF = 1.14).

Conference papers:

- P. Qi, H. Liu, L. Seneviratne, and K. Althoefer, “Design, kinematics and prototype of a flexible robot arm with planar springs,” in *Proc. ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE’15)*, Boston, USA, 2015, Paper DETC2015-46503.
- P. Qi, C. Liu, L. Zhang, S. Wang, H. K. Lam, and K. Althoefer, “Fuzzy logic control of a continuum manipulator for surgical applications,” in *Proc. IEEE International Conference on Robotics and Biomimetics (ROBIO’14)*, Bali, Indonesia, 2014.
- P. Qi, C. Qiu, H. Liu, J. S. Dai, L. Seneviratne, and K. Althoefer, “A novel continuum-style robot with multilayer compliant modules,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’14)*, Chicago, USA, 2014.
- P. Qi, H. Liu, L. Seneviratne, and K. Althoefer, “Towards kinematic modelling of a multi-DOF tendon driven robotic catheter,” *in Proc. 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC’14)*, Chicago, USA, 2014.
- G. Bai, P. Qi, K. Althoefer, D. Li, X. Kong, and J. S. Dai, “Kinematic analysis of a mechanism with dual remote centre of motion and its potential application,” in *Proc. ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE’15)*, Boston, USA, 2015.
- Y. Yu, P. Qi, K. Althoefer, and H. K. Lam, “Lagrangian dynamics and nonlinear control of a continuum manipulator,” in *Proc. IEEE International Conference on Robotics and Biomimetics (ROBIO’15)*, Zhuhai, China, 2015.
- C. Qiu, P. Qi, H. Liu, K. Althoefer, and J. S. Dai, “Six dimensional compliance analysis of ortho-planar springs for a continuum manipulator,” in *Proc. ASME 2014 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC’14)*, Buffalo, USA, 2014.

1.5 Thesis overview

The remainder of this thesis is organized as follows:

Chapter 2 presents a description of a concise overview of the mathematical frameworks used for mechanics analysis of the continuum manipulator. Firstly the coordinate transformations of rigid body motions are described. Then, the theory of screws based on linear algebra and matrix groups is presented; the concepts of twists and wrenches in screw coordinates are introduced; transformations of twists and wrenches between coordinate frames are described. These mathematical tools will particularly be used for six-dimensional compliance matrix derivation in Chapter 4. At last, differential geometry of curves are presented with an emphasis on the Frenet-Serret frame.

Chapter 3 presents the design concept of the proposed continuum manipulator with multilayer planar springs. Firstly a survey of frequently applied continuum-style robot constructions to date is presented and they are categorised according to the distinctive backbone architectures. Then, a description of the standard radial-leg design of the planar spring is presented, followed by the conceptual designs of the double-layer module and continuum manipulator assembly. Also, design variations of planar springs by varying the angle of leg arrangement are discussed. Next, experimental tests regarding the linear and decoupling characteristics of the out-of-plane rotation and translational motion of modules and the continuum manipulator are reported. Finally, discussions are provided to elucidate the advantages of the continuum manipulator and the potentials to improve the design in the future.

Chapter 4 presents the mechanics of the proposed continuum manipulator. Firstly the compliance characteristics of the planar spring are comprehensively investigated using a compliance matrix based approach under the framework of screw theory; this compliance matrix is derived with a bottom-up approach and reveals to be diagonal; FEA simulations are conducted to validate the main compliance elements of the compliance matrix. Then, an analysis regarding the flexibility of the fully assembled continuum manipulator is provided; two factors leading to the bending deflection decoupled with contraction are discussed; an overview of the continuum manipulator kinematics and statics is presented. Then, the constant-curvature approximation is utilised to derive the kinematics for both three-tendon-driven and single-tendon-driven continuum manipulators, laying the basis for kinematic control in the next chapter. At last, the dynamics of the continuum manipulator is studied based on the Lagrangian method.

Chapter 5 presents two fuzzy control strategies for continuum manipulators. Firstly a fuzzy logic controller is implemented on the single-tendon-driven continuum manipulator to autonomously execute the set-point regulation task in 3D space. MATLAB simulations are conducted to verify the feasibility of the fuzzy controller. Secondly a fuzzy-model-based controller is implemented to control the three-tendon-driven continuum manipulator to track a trajectory in 3D space, and the kinematic model in this case is

the most popular and widely used in the area of continuum robotics. Both MATLAB simulations and experiments are conducted to verify the superior performances of this controller compared to other classical controllers found in the literature to control the continuum manipulator.

Chapter 6 concludes with a summary of the research presented in this thesis and discusses potential future research and applications in continuum robotics.

Chapter 2 Mathematical Framework

Although continuum manipulators fundamentally differ from conventional rigid-link robots, the basic approaches to develop their (kinematic and dynamic) models are strongly correlated to those classical kinematics and mechanics principles. In this chapter, the theorems of the motion of rigid bodies and the coordinate transformations are rigorously presented based on Lie group, Lie algebra and screw theory, and such mathematical tools form the basis for describing the spatial continuum geometry and deformation in Chapter 4. There are two main parts in this chapter. First, the theoretical notions to represent rigid body kinematics and statics using screws, twists and wrenches in the form of one element of a six-dimensional vector space are introduced. Screw theory (Ball, 1900), (Brand, 1949), (Dai, 2014) provides a geometrical description of spatial rigid body motion and a compact representation for kinematics and statics. The research on the connection between screw theory and Lie algebra dates back to Klein's work in the 19^{th} century (Klein, 1872). The most recent research on this issue can be seen from Dai's book on screw algebra and Lie groups and Lie algebras (Dai, 2014), and his paper which reveals the intrinsic connection among Lie groups, Lie algebras and quaternions (Dai, 2015). This formulation integrating screw theory with Lie group and Lie algebra enables an efficient way to the analysis of motions and forces in manipulators, which will be seen in simplifying the compliance analysis of planar springs that comprise the proposed continuum manipulator. Second, the rest of this chapter provides a short introduction to the theory of curves, and the emphasis will be placed on the Frenet-Serret (F-S) frame transforms and the related differential geometry. These will contribute to the following derivations of kinematics and dynamics of the continuum manipulator.

2.1 Coordinate transformations in $\mathbb{E}^3$

2.1.1 Vectors

In three-dimensional Euclidean space (or $\mathbb{E}^3$), position of a point $\boldsymbol{p}$ is defined by using the triple $(p_x, p_y, p_z) \in \mathbb{R}^3$, and it is commonly represented in the form of a column vector $\left[p_x, p_y, p_z \right]^T$. This vector uniquely specifies a position viewed from a predefined right-handed orthogonal coordinate frame {*xyz*}, thus it is also called position vector which starts from the origin of frame {*xyz*} to point $\boldsymbol{p}$. Given another point $\boldsymbol{q}$, represented by $\left[q_x, q_y, q_z \right]^T$, in the same frame {*xyz*}, the directed line segment from point $\boldsymbol{p}$ to point $\boldsymbol{q}$ can be denoted by a vector $\boldsymbol{v} = \boldsymbol{q} - \boldsymbol{p}$ in coordinates. This vector $\boldsymbol{v}$ does not specify a unique physical entity but can connect any other two points that preserve the relative distance and direction. Thus, such vector $\boldsymbol{v} \in \mathbb{R}^3$ is the other type of vector, sometimes called free vector that can be located anywhere in space. In the following, all vectors mentioned later will refer to free vector unless indicated otherwise; the position vector will be termed as a point.

If using homogeneous coordinates (Baer, 2005) in $\mathbb{R}^4$, the presentation of point $\boldsymbol{p}$ is appended with an entry 1, leading to the form $\bar{\boldsymbol{p}} = \left[q_x, q_y, q_z, 1 \right]^T$, while the free vector $\boldsymbol{v}$ is appended with 0 as $\bar{\boldsymbol{v}} = \left[v_x, v_y, v_z, 0 \right]^T$. In this convention, it is obvious that the action of a vector with vector addition applied on a point results in a translation to another point. Further, based on the definition of vector, the addition of two free vectors gives another free vector; the addition of a free vector and a position vector gives a position vector; the addition of two position vectors makes no sense (Dai, 2014). Given two free vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$, their cross product $\boldsymbol{a} \times \boldsymbol{b}$ can be implemented through matrix multiplication where vector $\boldsymbol{a}$ is written in the form of skew-symmetric matrix as

$$\hat{\boldsymbol{a}} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

In this equation $\hat{\boldsymbol{a}}$ denotes the skew-symmetric matrix of vector $\boldsymbol{a}$ with the coordinates (a_x, a_y, a_z). The complete collection of matrix $\hat{\boldsymbol{a}}$ in the field of reals forms a Lie algebra *so*(3) which is the tangent space of Lie group *SO*(3) at the identity. This will be further described in 2.2.

2.1.2 Rotation matrices as *SO*(3) and its Lie algebra

The rotation matrices can be defined by giving the relative orientation between different coordinate frames (Tsai, 1999). Assuming that frame {*uvw*} is a different coordinate frame from frame {*xyz*}, the coordinate axes of frame {*uvw*} relative to frame {*xyz*} are described by vectors $\boldsymbol{u}_x$, $\boldsymbol{v}_y$, $\boldsymbol{w}_z$, then the rotation matrix representing the relative orientation between the two frames is defined by the 3×3 matrix $\mathbf{R} = \left[\boldsymbol{u}_x, \boldsymbol{u}_y, \boldsymbol{u}_z \right]$.

An action of such rotational transformation on a vector $\boldsymbol{v}$ will change the direction of that represented line segment. The action of a rotation matrix can also be treated as an orientation change of the original frame {*xyz*} that leads to a new frame.

All of the rotation matrices $\mathbf{R} \in \mathbb{R}^{3\times3}$ constructed in this manner have the following properties:

1. $\mathbf{R} \in \mathbb{R}^{3\times3}$ is an orthogonal matrix, satisfying $\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}$; here $\mathbf{I}$ represents a 3×3 identity matrix; thus $\mathbf{R}^{-1} = \mathbf{R}^T$;
2. The determinant of $\mathbf{R}$, i.e. $\det(\mathbf{R}) = 1$.

The set of all 3×3 matrices that possess the above two properties is called the special orthogonal group in three dimensions (Dai, 2014), (Murray, Li, & Sastry, 1994), written as

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3\times3} \mid \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = 1 \right\} \subset \mathbb{R}^{3\times3}$$

$SO(3) \subset \mathbb{R}^{3\times3}$ is a continuous group (Helgason, 1979) under the operation of matrix multiplication and it satisfies the following axioms:

1. Closure: if $\mathbf{R}_1, \mathbf{R}_2 \in SO(3)$, then $\mathbf{R}_1\mathbf{R}_2 \in SO(3)$;
2. Identity: the identity element in $SO(3)$ is the 3×3 identity matrix;
3. Inverse: for each $\mathbf{R} \in SO(3)$, its inverse $\mathbf{R}^{-1} = \mathbf{R}^T \in SO(3)$;
4. Associativity: for $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3 \in SO(3)$, it has $(\mathbf{R}_1\mathbf{R}_2)\,\mathbf{R}_3 = \mathbf{R}_1\,(\mathbf{R}_2\mathbf{R}_3)$.

The following elementary rotation matrices are mostly used throughout this thesis. They are performed about each of the three coordinate axes, respectively.

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$\mathbf{R}_y(\beta)=\begin{bmatrix}\cos\beta & 0 & \sin\beta\\ 0 & 1 & 0\\ -\sin\beta & 0 & \cos\beta\end{bmatrix}$$

$$\mathbf{R}_z(\phi)=\begin{bmatrix}\cos\phi & -\sin\phi & 0\\ \sin\phi & \cos\phi & 0\\ 0 & 0 & 1\end{bmatrix}$$

Note that rotation matrices can be combined; but when applying sequential transformations, the order of multiplications matters (Dai, 2014). The rotation about an arbitrary axis in space is discussed in (Dai, 2006), (Dai, 2015), and the derived rotation formula is called *Rodrigues' formula*.

To bring up the Lie algebra of *SO*(3), we consider the first property of rotation matrices. If **R** is a function of time *t*, taking the differential of **R** with respective to *t* leads to $\dot{\mathbf{R}}\mathbf{R}^T+\mathbf{R}\dot{\mathbf{R}}^T=\mathbf{0}$. We further have $\dot{\mathbf{R}}\mathbf{R}^T=-\mathbf{R}\dot{\mathbf{R}}^T=-(\dot{\mathbf{R}}\mathbf{R}^T)^T$ and recognize that $\dot{\mathbf{R}}\mathbf{R}^T$ and $\mathbf{R}\dot{\mathbf{R}}^T$ are skew-symmetric matrices, which are defined as

$$\mathbf{A}=\begin{bmatrix}0 & -a_z & a_y\\ a_z & 0 & -a_x\\ -a_y & a_x & 0\end{bmatrix}=\hat{\boldsymbol{a}}$$

where $\mathbf{A}=-\mathbf{A}^T\in\mathbb{R}^{3\times3}$ is skew-symmetric matrix and $\boldsymbol{a}=\left[a_x,a_y,a_z\right]^T\in\mathbb{R}^3$ is a vector. This equation actually defines a bijective mapping from $\mathbb{R}^3$ to the vector space comprised by the set of all 3×3 skew-symmetric matrices, using the $\wedge$ symbol, and the inverse operation is using the $\vee$ symbol, such that $\hat{\boldsymbol{a}}^{\vee}=\boldsymbol{a}$.

The set of all 3×3 skew-symmetric matrices is denoted *so*(3) (Murray, Li, & Sastry, 1994) and it is defined as

$$so(3)=\left\{\mathbf{A}\in\mathbb{R}^{3\times3}\mid\mathbf{A}^T=-\mathbf{A}\right\}\subset\mathbb{R}^{3\times3}$$

The set *so*(3) is a three-dimensional vector space, which represents the tangent space of *SO*(3) at its identity and is called the Lie algebra of *SO*(3). It satisfies the following properties:

1. $\mathbf{A},\mathbf{B}\in so(3)$, then $\mathbf{A}+\mathbf{B}\in so(3)$;
2. If θ is a scalar, and $\mathbf{A}\in so(3)$, then $\theta\mathbf{A}\in so(3)$;
3. If $\hat{\boldsymbol{a}},\hat{\boldsymbol{b}}\in so(3)$, then the relation $\left(\boldsymbol{a}+\boldsymbol{b}\right)^{\wedge}=\hat{\boldsymbol{a}}+\hat{\boldsymbol{b}}$ holds.

Usually, it is convenient to define a unit skew-symmetric matrix that holds $\hat{\boldsymbol{s}}\in so(3),|\,\hat{\boldsymbol{s}}\,|=\mathbf{I}$. Referring to equation $\dot{\mathbf{R}}\mathbf{R}^T+\mathbf{R}\dot{\mathbf{R}}^T=\mathbf{0}$, we can write it as $\hat{\boldsymbol{\omega}}+\hat{\boldsymbol{\omega}}^T=\mathbf{0}$, where $\boldsymbol{\omega}=\dot{\theta}\boldsymbol{s}$, $\boldsymbol{s}$ denotes the rotation axis and

later will be defined as a screw axis, $\hat{\boldsymbol{s}}$ is a unit skew-symmetric matrix; $\dot{\theta}$ denotes the angular velocity about the rotation axis ***s***.

2.1.3 Homogenous transformations

A transformation between two coordinate frames can be represented by g = (***d***, **R**), consisting of a position vector $\boldsymbol{d} \in \mathbb{R}^3$ and a rotation matrix $\mathbf{R} \in SO(3)$ (Selig J. M., 2004). The action of this transformation changes the coordinates of a point in one frame into the coordinates in another frame. The semi-direct product of $\mathbb{R}^3$ with $SO(3)$ gives the special Euclidean group, which is denoted as $SE(3)$. $SE(3)$, also known as rigid motion group, is a six-dimensional smooth manifold with a group structure. It is defined as

$$SE(3) = \left\{ g = (\boldsymbol{d}, \mathbf{R}) \mid \boldsymbol{d} \in \mathbb{R}^3, \mathbf{R} \in SO(3) \right\} = \mathbb{R}^3 \propto SO(3)$$

The homogenous representation of g is written in the form of

$$\overline{g} = \begin{bmatrix} \mathbf{R} & \boldsymbol{d} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$$

Here, ***0*** presents a 3×1 zero vector, i.e. $\boldsymbol{0}^T = [0, 0, 0]$. And the inverse of g is given by

$$\overline{g}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T \boldsymbol{d} \\ \boldsymbol{0}^T & 1 \end{bmatrix}$$

The action of a homogenous transformation on a point ***p*** in homogeneous coordinates is represented as

$$\overline{g} \cdot \overline{\boldsymbol{p}} = \begin{bmatrix} \mathbf{R} & \boldsymbol{d} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}\boldsymbol{p} + \boldsymbol{d} \\ 1 \end{bmatrix}$$

which takes the point ***p*** to a new position; while this action on a vector is as

$$\overline{g} \cdot \overline{\boldsymbol{v}} = \begin{bmatrix} \mathbf{R} & \boldsymbol{d} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}\boldsymbol{v} \\ 0 \end{bmatrix}$$

where we can see that the translation does not lead to any changes as expected.

An element of $SE(3)$ thus can represent a general rigid body motion that involves rotation and translation, in which case $\boldsymbol{d} \in \mathbb{R}^3$ and $\mathbf{R} \in SO(3)$ are called displacement operators and compulsorily they are represented in the same coordinate frame (Dai, 2014), (Murray, Li, & Sastry, 1994), (Selig J. M., 2004).

2.2 Screw motion

Chasles' theorem (Kumar), (Dai, 2006)

"*The most general rigid body displacement can be produced by a translation along a line followed (or preceded) by a rotation about that line*."

This displacement in *Chasles' theorem* is called a screw motion, which is modelled by the curve around an axis $\boldsymbol{s}$ (please refer to Figure 2-1). This screw axis $\boldsymbol{s}$ is specified by a position vector $\boldsymbol{r}_O$ in the coordinate from {*xyz*} and the vector $\boldsymbol{r}_O$ is perpendicular with axis $\boldsymbol{s}$. The infinitesimal version of a screw motion is called a twist $\boldsymbol{T}$ which can be presented in the form of a unit screw $\boldsymbol{S}$ multiplied by a scalar ω as $\boldsymbol{T}=\omega\boldsymbol{S}$. Twist $\boldsymbol{T}$ represented by a unit screw $\boldsymbol{S}$ describes the instantaneous velocity of a point P on a rigid body that consists of an angular velocity $\omega\boldsymbol{s}$ around an axis and a linear velocity $h\boldsymbol{s}$ along this axis. This can be illustrated in Figure 2-1 and shall be discussed further in the following context of this section.

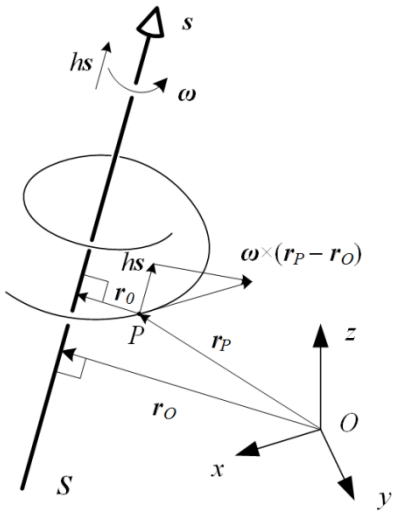


Figure 2-1 A generalized motion of a rigid body in screw representation.

2.2.1 Twists

In the preceding section, we discussed the rigid body motion. A movement of a rigid body can be parameterised by a spatial displacement. Based on *Chasles' theorem*, any finite displacement (Dai, Holland, & Kerr, 1995), (Huang & Roth, 1994), (Dai, 2012) can be equivalent to a translation vector $\boldsymbol{d} \in \mathbb{R}^3$ and a rotation matrix $\mathbf{R} \in SO(3)$, whose rotation axis aligns with the direction of the translation vector $\boldsymbol{d}$. Thus, a point $\boldsymbol{p}$ in a moving rigid body tracks the curve $\boldsymbol{p}(t)$ represented by

$$\boldsymbol{p}(t) = \mathbf{R}(t)\boldsymbol{p} + \boldsymbol{d}(t) \tag{2-1}$$

Then, the instantaneous velocity of the point $\boldsymbol{p}$ is derived by taking the time differential of the above equation, and we get

$$\dot{\boldsymbol{p}}(t) = \dot{\mathbf{R}}(t)\boldsymbol{p} + \dot{\boldsymbol{d}}(t) \tag{2-2}$$

Now substitute $\boldsymbol{p} = \mathbf{R}^T(t)\big(\boldsymbol{p}(t) - \boldsymbol{d}(t)\big)$ (which is derived from (2-1)) into the above equation to obtain

$$\dot{\boldsymbol{p}}(t) = \dot{\mathbf{R}}(t)\mathbf{R}^T(t)\big(\boldsymbol{p}(t) - \boldsymbol{d}(t)\big) + \dot{\boldsymbol{d}}(t) \tag{2-3}$$

The term $\dot{\mathbf{R}}(t)\mathbf{R}^T(t)$ in the above equation is a skew-symmetric matrix which we have discussed before, and conventionally we used $\hat{\boldsymbol{\omega}}(t)$ to represent this term. In such a way $\boldsymbol{\omega}(t)$ can be used to represent angular velocities that are associated with the rotation axis and rotation angles (Dai, 2014). Then (2-3) can be rewritten as

$$\dot{\boldsymbol{p}}(t) = \boldsymbol{\omega}(t) \times \boldsymbol{p}(t) - \boldsymbol{\omega}(t) \times \boldsymbol{d}(t) + \dot{\boldsymbol{d}}(t) \tag{2-4}$$

Define a 4×4 matrix $\hat{\boldsymbol{\xi}}$ to be $\hat{\boldsymbol{\xi}} = \begin{bmatrix} \hat{\boldsymbol{\omega}}(t) & \boldsymbol{v}(t) \\ \boldsymbol{0}^T & 0 \end{bmatrix} \in \mathbb{R}^{4\times4}$, with $\boldsymbol{v}(t) = -\boldsymbol{\omega}(t) \times \boldsymbol{d}(t) + \dot{\boldsymbol{d}}(t)$, then (2-4) can be rewritten in homogenous coordinates as

$$\begin{bmatrix} \dot{\boldsymbol{p}}(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{R}}(t)\mathbf{R}^T(t) & -\dot{\mathbf{R}}(t)\mathbf{R}^T(t) \times \boldsymbol{d}(t) + \dot{\boldsymbol{d}}(t) \\ \boldsymbol{0}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\omega}}(t) & \boldsymbol{v}(t) \\ \boldsymbol{0}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix} \tag{2-5}$$

from which we can obtain $\dot{\bar{\boldsymbol{p}}}(t) = \hat{\boldsymbol{\xi}}\,\bar{\boldsymbol{p}}$.

Analogous to the definition of *so*(3) but focusing on the generalized velocity, we can define

$$se(3) = \left\{ (\boldsymbol{v}, \hat{\boldsymbol{\omega}}) \mid \boldsymbol{v} \in \mathbb{R}^3, \hat{\boldsymbol{\omega}} \in so(3) \right\}$$

An element of *se*(3) is referred to as a twist. We take the inverse operation of the representation $\hat{\boldsymbol{\xi}} \in se(3)$ and derive the six-dimensional vector as

$$\begin{bmatrix} \hat{\boldsymbol{\omega}}(t) & \boldsymbol{v}(t) \\ \boldsymbol{0}^T & 0 \end{bmatrix}^{\vee} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix}$$

and further the twist coordinates of $\hat{\boldsymbol{\xi}}$ is defined as $\boldsymbol{\xi} \coloneqq (\boldsymbol{v}, \boldsymbol{\omega}) \in \mathbb{R}^6$.

A screw $\boldsymbol{S}$ can be specified by an axis $\boldsymbol{s}$, a position vector $\boldsymbol{r}_o$, and a pitch h. Recall that the component $\boldsymbol{\omega} = \dot{\theta}\boldsymbol{s}$, and similarly, we define the linear velocity $\dot{d}$ satisfying $\dot{\boldsymbol{d}} = \dot{d}\boldsymbol{s}$, then the twist $\boldsymbol{\xi} = (\boldsymbol{v}, \boldsymbol{\omega})$ ($\boldsymbol{\omega} \neq \boldsymbol{0}$) is associated with a screw having the following attributed (Hunt, 1990):

1. Pitch $h = \dfrac{\boldsymbol{\omega}\boldsymbol{v}}{\|\boldsymbol{\omega}\|^2} = \dfrac{\dot{d}}{\dot{\theta}}$;

2. Axis $\boldsymbol{s} = \dfrac{\boldsymbol{\omega} \times \boldsymbol{v}}{\|\boldsymbol{\omega}\|^2}$;

3. Magnitude of the twist $m = \|\boldsymbol{\omega}\|^2$.

Therefore, a twist $\boldsymbol{\xi} = (\boldsymbol{v}, \boldsymbol{\omega})$ ($\boldsymbol{\omega} \neq \boldsymbol{0}$) defined above is a screw multiplied by the magnitude $\|\boldsymbol{\omega}\|$, i.e. $\boldsymbol{\xi} = \|\boldsymbol{\omega}\|\boldsymbol{s}$. Since a screw uniquely corresponds to a point on the Klein quadric in five-dimensional projective space $\mathbb{P}^6$, it has five independent parameters. Attaching to a screw a parameter which indicates the amplitude, a twist can be obtained and in this case we can see the twist derived from the screw has six independent parameters.

2.2.2 Coordinate transformations in *se*(3)

In this section, we will discuss how an element of *se*(3) is transformed from a coordinate frame to another. We employ the twist belonging to *se*(3) to show the procedure of performing the coordinate transformation in *se*(3) and this naturally applies to the dual Lie algebra *se**(3) constituted by all wrenches.

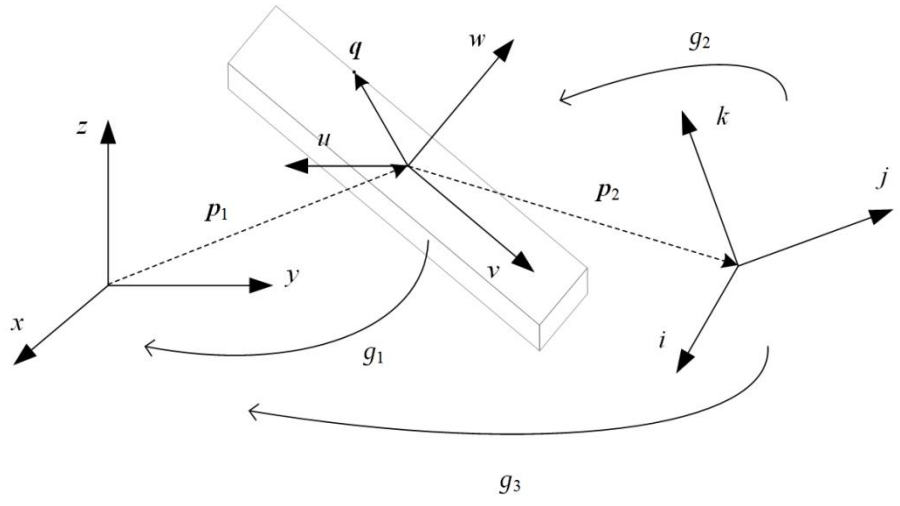


Figure 2-2 Coordiante frames for specifying twists.

In Figure 2-2, coordinate frame {*xyz*} is considered as a fixed frame, and coordinate frame {*uvw*} is attached to a rigid body. Frame {*uvw*} relative to frame {*xyz*} can be represented by an element g_1 of the group $SE(3)$ that consists of a translation vector $\boldsymbol{p}_1 \in \mathbb{R}^3$ and a rotation matrix $\mathbf{R}_1 \in SO(3)$, which is discussed in 2.1.3. The transformation $g_1 \in SE(3)$ is written in homogenous coordinates as

$$g_1 = \begin{bmatrix} \mathbf{R}_1 & \boldsymbol{p}_1 \\ \boldsymbol{0}^T & 1 \end{bmatrix} \tag{2-6}$$

A point $\boldsymbol{q}$ is represented in frame {*uvw*} as a position vector, and we can derive $\boldsymbol{q}'$ as the position of point $\boldsymbol{q}$ viewed from frame {*xyz*} by a transformation of coordinates

$$\begin{bmatrix} \boldsymbol{q}' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 & \boldsymbol{p}_1 \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ 1 \end{bmatrix} \tag{2-7}$$

Considering the rigid body tracing a trajectory with $g_1(t) \in SE(3)$ parameterised by time *t*, (2-7) can be rewritten as

$$\begin{bmatrix} \boldsymbol{q}'(t) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1(t) & \boldsymbol{p}_1(t) \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ 1 \end{bmatrix} \tag{2-8}$$

and thus we can have

$$\dot{\boldsymbol{q}}'(t) = \dot{\mathbf{R}}_1(t)\boldsymbol{q} + \dot{\boldsymbol{p}}_1(t) \tag{2-9}$$

where $\dot{\boldsymbol{q}}'(t)$ denotes the spatial velocity of point $\boldsymbol{q}$ viewed from frame {*xyz*}. However, the vector $\boldsymbol{q}$ still represents coordinates in frame {*uvw*}. To this end, we will rewrite the equation as

$$\begin{aligned} \dot{\boldsymbol{q}}'(t) &= \dot{\mathbf{R}}_1(t)\mathbf{R}_1^{-1}(t)\underbrace{\mathbf{R}_1(t)\boldsymbol{q}}_{=\boldsymbol{q}'-\boldsymbol{p}_1(t)} + \dot{\boldsymbol{p}}_1(t) \\ \text{i.e.} \quad \dot{\boldsymbol{q}}'(t) &= \dot{\mathbf{R}}_1(t)\mathbf{R}_1^{-1}(t)\boldsymbol{q}' - \dot{\mathbf{R}}_1(t)\mathbf{R}_1^{-1}(t)\boldsymbol{p}_1(t) + \dot{\boldsymbol{p}}_1(t) \end{aligned} \tag{2-10}$$

This representation is in line with the form of a twist, and we define the instantaneous spatial angular velocity as

$$\hat{\boldsymbol{\omega}}_1^s(t) = \dot{\mathbf{R}}_1(t)\mathbf{R}_1^{-1}(t)$$

and the instantaneous spatial linear velocity as

$$\boldsymbol{v}_1^s(t) = -\hat{\boldsymbol{\omega}}_1^s(t)\boldsymbol{p}_1(t) + \dot{\boldsymbol{p}}_1(t)$$

Therefore, (2-10) can be rewritten in the form as

$$\begin{bmatrix} \dot{\boldsymbol{q}}'(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{R}}_1(t)\mathbf{R}_1^{-1}(t) & -\dot{\mathbf{R}}_1(t)\mathbf{R}_1^{-1}(t)\boldsymbol{p}_1(t) + \dot{\boldsymbol{p}}_1(t) \\ \boldsymbol{0}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\omega}}_1^s(t) & \boldsymbol{v}_1^s(t) \\ \boldsymbol{0}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{q}' \\ 1 \end{bmatrix}$$

Here we define $\hat{V}_1^s(t) = \begin{bmatrix} \hat{\boldsymbol{\omega}}_1^s(t) & \boldsymbol{v}_1^s(t) \\ \boldsymbol{0}^T & 0 \end{bmatrix}$ as the instantaneous spatial velocity, which is exactly a twist to describe the instantaneous velocity of the rigid body viewed from frame $\{xyz\}$. In twist coordinates, we have $V_1^s(t) = \begin{bmatrix} \boldsymbol{v}_1^s(t) \\ \boldsymbol{\omega}_1^s(t) \end{bmatrix}$. Further, the following relationship exists $\hat{V}_1^s = \dot{g}_1 g_1^{-1}$.

Similarly, we can derive the motion equation of the rigid body viewed form frame $\{uvw\}$

$$\begin{bmatrix} \dot{\boldsymbol{q}}(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1^{-1}(t)\dot{\mathbf{R}}_1(t) & \mathbf{R}_1^{-1}(t)\dot{\boldsymbol{p}}_1(t) \\ \boldsymbol{0}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\omega}}_1^b(t) & \boldsymbol{v}_1^b(t) \\ \boldsymbol{0}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ 1 \end{bmatrix}$$

Here we define the spatial velocity of the rigid body with respect to the instantaneous body frame, denoted by

$$\hat{V}_1^b(t) = \begin{bmatrix} \hat{\boldsymbol{\omega}}_1^b(t) & \boldsymbol{v}_1^b(t) \\ \boldsymbol{0}^T & 0 \end{bmatrix}$$

Further, the relationship $\hat{V}_1^b = g_1^{-1}\dot{g}_1$ exists. Therefore, we can find the relationship

$$\hat{V}_1^s = g_1 g_1^{-1}\dot{g}_1 g_1^{-1} = g_1 \hat{V}_1^b g_1^{-1} \tag{2-11}$$

and converting it to twist coordinates,

$$\begin{bmatrix} \boldsymbol{v}_1^s \\ \boldsymbol{\omega}_1^s \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 & \hat{\boldsymbol{p}}_1\mathbf{R}_1 \\ \boldsymbol{0}^T & \mathbf{R}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1^b \\ \boldsymbol{\omega}_1^b \end{bmatrix} \tag{2-12}$$
$$\Leftrightarrow V_1^s = \mathrm{Ad}_g V_1^b$$

Based on the above equation, we can see that the 6×6 matrix Ad_g transforms twists from one coordinate frame to another. The matrix Ad_g is known as the adjoint transformation of $g \in SE(3)$. In the above case, the matrix Ad_g maps twist V_1^b to twist V_1^s, thus, $\mathrm{Ad}_g : \mathbb{R}^6 \to \mathbb{R}^6$. When the twist is presented in a matrix form, the adjoint action can be completed by the conjugation of Lie group $SE(3)$. This conjugation action and operation is shown in (2-11). When the twist is presented in an equivalent vector form, from (2-12) we can see the adjoint action can be completed by the left action of Lie group $SE(3)$. The two manners of implementing the adjoint action of $SE(3)$ on $se(3)$ are equivalent.

In the following section, the adjoint transformation shall play an important role in dealing with the transformation of twists and wrench. Here, the inverse and the inverse transpose of such transformation matrix are given respectively by

$$\mathrm{Ad}_g^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\hat{\boldsymbol{p}} \\ \mathbf{0} & \mathbf{R}^T \end{bmatrix}; \qquad \mathrm{Ad}_g^{-T} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \hat{\boldsymbol{p}}\mathbf{R} & \mathbf{R} \end{bmatrix}$$

The velocities can be transformed between coordinate frames. Now consider the motion of three coordinate frames, {*xyz*}, {*uvw*}, and {*ijk*}. Regarding the spatial velocities, the following relationship holds

$$V_3^s = V_1^s + \mathrm{Ad}_{g_1} V_2^s \tag{2-13}$$

where V_1^s, V_2^s, V_3^s are the spatial velocities of frame {*uvw*} relative to frame {*xyz*}, frame {*ijk*} relative to frame {*vuw*}, frame {*ijk*} relative to frame {*xyz*}, respectively; $g_1 \in SE(3)$ denote a rigid body transformation, and thus Ad_{g_1} accordingly maps coordinate frame {*uvw*} into frame {*xyz*}.

Regarding the body velocities, similarly, we can derive the following relationship of transformation,

$$V_3^b = \mathrm{Ad}_{g_2}^{-1} V_1^b + V_2^b \tag{2-14}$$

where V_3^b is the body velocity of frame {*ijk*} relative to frame {*xyz*}, but viewed from the body frame itself. And V_1^b, V_2^b are similarly defined body velocities as well. For a detailed proof, please refer to (Murray, Li, & Sastry, 1994).

Most often, two of the coordinate frames are stationary with respect to each other, and thus (2-13) can be simplified into

$$V_3^s = \mathrm{Ad}_{g_1} V_2^s \tag{2-15}$$

We will make the frequent use of the above equation in this dissertation.

Because body velocities are always viewed from the body frame, thus we have the relation: $V_3^b = V_2^b$. This is easy to be interpreted as the independency of the local body velocity.

2.3 Wrenches

Poinsot's theorem (Hunt, 1990), (Dai, 2006)

"*Any system of forces acting on a rigid body can be replaced by a single force applied along a line, combined with a torque about that same line.*"

This generalized force is called wrench and it represents a force/moment pair. We write this wrench as a vector in $\mathbb{R}^6$:

$$\mathbf{W} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{m} \end{bmatrix} \in \mathbb{R}^6$$

where force $\boldsymbol{f} \in \mathbb{R}^3$ represents the linear component and moment $\boldsymbol{m} \in \mathbb{R}^3$ represents the angular component. All feasible wrenches constitutes an space dual with the space of twists *se*(3). Hence the collection of wrenches is called the dual Lie algebra *se**(3). The duality can be represented by taking different types of coordinates for twists and wrenches, which is the well-known ray and axis coordinates in theoretical kinematics. This will be discussed in the derivation of the compliance matrix of the proposed continuum manipulator in Chapter 4

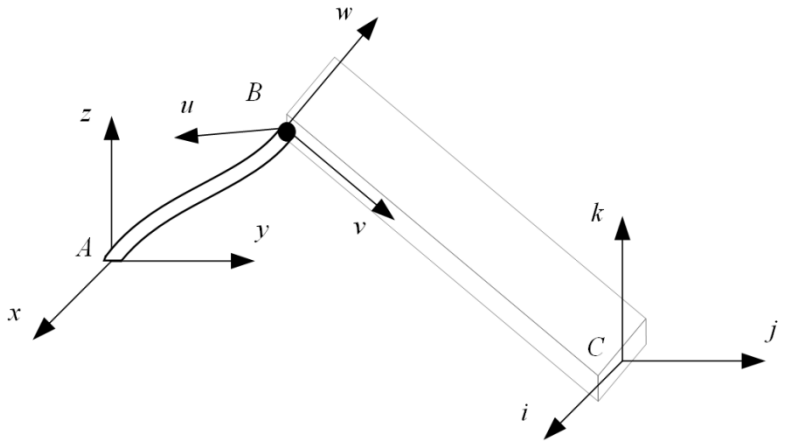


Figure 2-3 Transformation of wrenches applied on the rigid body between coordiante frames.

Consider frame {*vuw*} is attached to a rigid body, and a wrench applied at the origin of frame B, (please refer to Figure 2-3), is denoted as $\mathbf{W}_b = \left[\boldsymbol{f}_b, \boldsymbol{m}_b\right]^T \in \mathbb{R}^6$, whose entries are with respect to the local body frame {*vuw*}. Assume there is another coordinate frame {*ijk*} attached to the rigid body, we would like to derive an equivalent wrench $\mathbf{W}_c = \left[\boldsymbol{f}_c, \boldsymbol{m}_c\right]^T \in \mathbb{R}^6$ applied at the origin of frame {*ijk*}. Two wrenches are considered to be equivalent if they generate the same work for every possible rigid body motion. Now we write each of the virtual works, and assume they are equal,

$$\left(V_1^b\right)^T \mathbf{W}_b = \left(V_3^b\right)^T \mathbf{W}_c \tag{2-16}$$

where V_1^b, V_3^b are instantaneous body velocities with respect to an inertial frame {*xyz*}, as view from each local body frames.

Moreover, recall (2-14), in this case, $V_2^b = 0$ due to that frame {*vuw*} and frame {*ijk*} are stationary to each other in the rigid body, thus the following relationship holds $V_1^b = \mathrm{Ad}_{g_2} V_3^b$.

Therefore, (2-16) can be written as

$$\begin{aligned}
&\left(\mathrm{Ad}_{g_2} V_3^b\right)^T W_b = \left(V_3^b\right)^T W_c \\
&\Rightarrow \left(V_3^b\right)^T \mathrm{Ad}_{g_2}^T W_b = \left(V_3^b\right)^T W_c \\
&\Rightarrow \mathrm{Ad}_{g_2}^T W_b = W_c
\end{aligned} \tag{2-17}$$

Explicitly, the above equation can be rewritten as

$$\begin{bmatrix} \boldsymbol{f}_c \\ \boldsymbol{m}_c \end{bmatrix} = \begin{bmatrix} \mathbf{R}_2^T & \mathbf{0} \\ -\mathbf{R}_2^T \hat{\boldsymbol{p}}_2 & \mathbf{R}_2^T \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_b \\ \boldsymbol{m}_b \end{bmatrix}, \quad \left(\mathbf{Ad}_g^T = \begin{bmatrix} \mathbf{R}^T & \mathbf{0} \\ -\mathbf{R}^T \hat{p} & \mathbf{R}^T \end{bmatrix} \right)$$

In this case, the coordinate transformation of wrenches in the vector form is completed by the left action of an element of *SE*(3), which is analogous to the coordinate transformation of twists. Equation (2-17) will be made a frequently use when doing statics analysis in the following section.

2.4 Differential geometry of curves

This section will introduce the fundamental knowledge to analyse the geometrical curves which can be abstracted from the central backbone of the continuum manipulator. The differential geometry (Spivak, 1999) of curves will serve as the basis for one way to develop the kinematics and dynamics of the continuum manipulator in the following section. In this section, we will particularly focus on a type of coordinate system, i.e. Frenet-Serret (F-S) frame, which is moving along a continuous, differentiable curve in three-dimensional Euclidean space $\mathbb{E}^3$.

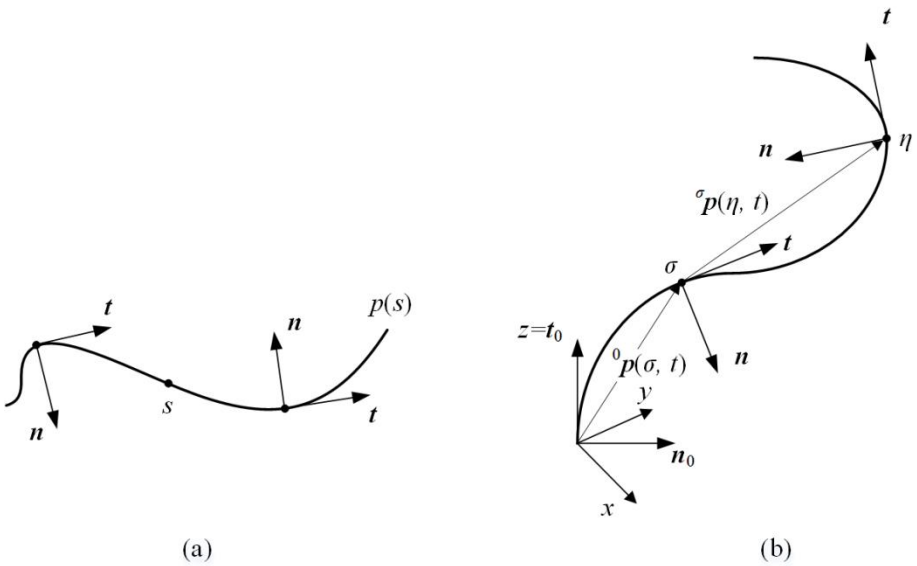


Figure 2-4 Frenet-Serret (F-S) frame along the curve in space. (a) a moving local F-S frame along the curve; (b) the F-S frame in a normal Cartesian coordinate frame {*xyz*} and the relationship between the local F-S frames.

Firstly, the curve $p(s)$ is parametrised by curve length s, $s \in [0, l]$, where l is the total length of the continuum manipulator backbone (please refer to Figure 2-4(a)). The F-S frame is constructed by three principle vectors, the unite vector $\boldsymbol{t}(s)$ tangent to the curve $p(s)$ at position s, the unit vector $\boldsymbol{n}(s)$ normal to the curve associated to $\boldsymbol{t}(s)$, and binormal unit vector $\boldsymbol{b}(s) = \boldsymbol{t}(s) \times \boldsymbol{n}(s)$ completing the right-hand frame. Therefore, the F-S frame is commonly called TNB frame and the three axes are usually often named as T, N, and B. We can see that this frame is a moving local frame along the curve to be studied. This curve is an ordinary curve that is not necessary to have a constant-curvature. The evolution of the curve are defined by the following *Frenet-Serret formulas*

$$\begin{aligned}
\frac{d\boldsymbol{t}(\mathrm{s})}{ds} &= k(\mathrm{s})\boldsymbol{n}(\mathrm{s}) \\
\frac{d\boldsymbol{n}(\mathrm{s})}{ds} &= -k(\mathrm{s})\boldsymbol{t}(\mathrm{s}) + \tau(\mathrm{s})\boldsymbol{b}(\mathrm{s}) \\
\frac{d\boldsymbol{b}(\mathrm{s})}{ds} &= -\tau(\mathrm{s})\boldsymbol{n}(\mathrm{s})
\end{aligned}$$

where $k(\mathrm{s})$ and $\tau(\mathrm{s})$ are the curvature and torsion respectively at each position s along the curve.

The *Frenet-Serret formulas* are also known as Frenet-Serret theorem which can be represented in form of matrix notation,

$$\begin{bmatrix} \boldsymbol{t}' \\ \boldsymbol{n}' \\ \boldsymbol{b}' \end{bmatrix} = \begin{bmatrix} 0 & k & 0 \\ -k & 0 & \tau \\ 0 & -\tau & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{t} \\ \boldsymbol{n} \\ \boldsymbol{b} \end{bmatrix}$$

where the prime indicates the differentiation with respect to s and for concision purpose, the parameter s is all omitted in matrix notation.

As seen in Figure 2-4(b), if simultaneously considering a normal Cartesian coordinate frame $\{xyz\}$, whose z-axis is aligned with the tangent to the curve and coincides with $\boldsymbol{t}(s)$, then in this formulation, for the planar curve, the orientation of the F-S frame $\{\mathrm{NBT}\}$ at position σ is given by

$$ {}^{0}\Phi(\sigma,t) = \mathbf{R}_{y}(\sigma k(\sigma,t)) = \begin{bmatrix} \cos(\sigma k(\sigma,t)) & 0 & \sin(\sigma k(\sigma,t)) \\ 0 & 1 & 0 \\ -\sin(\sigma k(\sigma,t)) & 0 & \cos(\sigma k(\sigma,t)) \end{bmatrix} \tag{2-18}$$

Here the front-superscript of ${}^{0}\Phi(\sigma, t) \in SO(3)$means that the orientation of the frame $\{\mathrm{N}_{\sigma}\mathrm{B}_{\sigma}\mathrm{T}_{\sigma}\}$ is represented relative to the base frame $\{\mathrm{N}_0\mathrm{B}_0\mathrm{T}_0\}$.

The change of the above orientation matrix along the curve thus can be parameterized by

$$\frac{\partial\, {}^{0}\Phi(\sigma,t)}{\partial\sigma} = {}^{0}\Phi(\sigma,t)\hat{\boldsymbol{a}}(\sigma,t) \tag{2-19}$$

where $\hat{\boldsymbol{a}}(\sigma,t) \in se(3)$ is a skew-symmetric matrix as discussed before, which is also can be treated as the produce of $\boldsymbol{a}(\sigma, t) \in \mathbb{R}^3$ after a linear operation. They are written as follows

$$\hat{\boldsymbol{a}}(\sigma,t) = \begin{bmatrix} 0 & 0 & k(\sigma,t) \\ 0 & 0 & 0 \\ -k(\sigma,t) & 0 & 0 \end{bmatrix}; \ \boldsymbol{a}(\sigma,t) = \begin{bmatrix} 0 \\ k(\sigma,t) \\ 0 \end{bmatrix}$$

In different coordinate formulations, they format of the above orientation matrix and its associated entities are varying in form.

Next, let ${}^0\boldsymbol{p}(\sigma, t) \in \mathbb{R}^3$ be the position vector at position σ along the curve viewed from the base frame $\{N_0B_0T_0\}$. This position vector can be derived based on the orientation matrix as

$${}^0\boldsymbol{p}(\sigma,\mathrm{t}) = \int_0^{\sigma} {}^0\Phi(\eta,\mathrm{t})\boldsymbol{e}_z d\eta \tag{2-20}$$

where $\boldsymbol{e}_z = [0,0,1]^T$ is the unit vector to specify the direction.

Moreover, we define the orientation matrix at point η related to ${}^0\Phi(\sigma, t)$ in the following relation,

$${}^0\Phi(\eta, t) = {}^0\Phi^T(\sigma, t)\,{}^0\Phi(\eta, t) \tag{2-21}$$

And thus, the positon vector ${}^{\sigma}\boldsymbol{p}(\eta, t) \in \mathbb{R}^3$ at position η viewed from the frame $\{N_\sigma B_\sigma T_\sigma\}$ is calculated as

$${}^{\sigma}\boldsymbol{p}(\eta, t) = {}^0\Phi^T(\sigma, t){}^0\boldsymbol{p}(\eta, t) \tag{2-22}$$

Furthermore, we have the special case holding as,

$${}^{\sigma}\boldsymbol{p}(\sigma, t) = {}^0\Phi^T(\sigma, t){}^0\boldsymbol{p}(\sigma, t) = -\,{}^0\boldsymbol{p}(\sigma, t) \tag{2-23}$$

So far, we have given the basis of differential geometry for a planar curve. When studying the kinematics and dynamics of a continuum manipulator in this thesis, we always assume there is no torsion along its backbone. The knowledge and formulas presented in this section will be a general case for the following constant-curve based derivation of the continuum manipulator mechanics.

Chapter 3 Design of Continuum Manipulators with Multilayer Planar Springs

This chapter introduces the design concept of the continuum manipulator with serially connected multiple layers of planar springs. Firstly, other continuum manipulator structures appeared in the context of the established works are reviewed. These robots are categorized based on their distinguished backbone architectures. This categorization aims to reveal the fundamental design principles underlying the popular continuum robots constructed to date. Then, a unique type of planar spring – ortho-planar spring, created by Parise and Howell et al. (Parise, Howell, & Magleby, 2001), is presented. We utilise both of the translational and rotational motions of the central platform of this spring to develop a double-layer module, and then assemble the modules into a flexible continuum manipulator. Next, design variations of planar springs are discussed and they are created by varying the layout of structural flexure beam elements. Integrating a series of different planar springs along the backbone, a continuum manipulator with a conic shape is thus developed.

The planar-spring-based continuum manipulator provides several notable advantages over existing designs. First, it possesses precise linear large-displacement motion. In this context, we utilise the linear output motion of each layer of springs. With the serial connection of multiple conjoined layers, the manipulator demonstrates linear bending even when executing large bends. Second, compared with work elsewhere, this structure demonstrates an effective way of decoupling bending from contraction and expansion. It reduces the uncontrolled compression when generating normal deflections and thus controlling robot bending is simplified. Third, the reachable workspace of the end-effector is enlarged by means of varying the length of the continuum manipulator via controlled contraction and expansion. A 3D-printed prototype of this continuum manipulator is experimentally evaluated. The conducted experiments show the validity of our approach.

3.1 Related works

A continuum manipulator is characterized by its continuously bending structure. Some hyper-redundant manipulators (Chirikjian, 1994) have, externally viewed, the appearance of a continuum arm; however, if they are internally comprised of a segmented backbone with many short rigid links/columns, they do not represent, strictly speaking, continuum manipulators – we will refer to such manipulators as "continuum-style manipulators". Different classifications of continuum-style robots have been described based on the location of mechanical actuator (Robinson & Davies, 1999) or on the backbone type and the actuation mode (Walker, 2013), (Webster & Jones, 2010). In this chapter, the frequently applied continuum-style robot constructions to date are then summarized according to their distinctive backbone architectures. Please refer to Figure 3-1. Particularly, this category also includes the subset of biologically inspired soft robots. Of these design principles, the earlier robot structure is composed of serially connected independent joints, and thus pertains to the aforementioned hyper-redundant manipulator. Examples include different kinds of snake-like robots, which are articulated by revolute joints (Hirose, 1993), universal joints (Buckingham, 2002), (Wright, et al., 2007) or spherical joints (Liao, Li, & Du, 2012). In terms of actuation mode, this type of design employs an individual micro-motor per joint (Wright, et al., 2007), or more commonly incorporates tendon-driven actuation (Buckingham, 2002), (Liao, Li, & Du, 2012). The designs share the advantages of having a large number of DOFs and accurate control, however they suffer from the problems of lighter payload, joint friction and incompressibility/inextensibility.

Perhaps the most common form of a truly continuum robot is the one that uses a spring backbone (Anderson & Horn, 1967), (Yang, Jason, & Abdel-Malek, 2006), (Mehling, Diftler, Chu, & Valvo, 2006), (Watanabe, Kanou, Kobayashi, & Fujie, 2011). Due to the flexibility of the spring structure, the shape of such a robot can be actuated in a tendon-driven manner and allows an ideal axial compliance combined with a relatively low hysteresis. Here, tendons are routed along the spring backbone producing torques at the termination points, which give rise to an under-actuated design. When forces are applied to the tendons, compression and bending deflections can be obtained. However, in such robots compression and bending deflections are mechanically coupled, leading to a bending actuation that results partially in compression (Walker, 2013). Similarly to spring backbone design, the bending flexure joint type is another form of constructing continuum manipulators and is often used in steerable MIS instruments (Jel ńek, Arkenbout, Henselmans, Pessers, & Breedveld, 2015). These robots are made of compliant materials and the flexural parts are carved from a single piece of thin-walled shell allowing bending. This structure essentially acts like springs but suffers from poor bending performance, which strongly depends

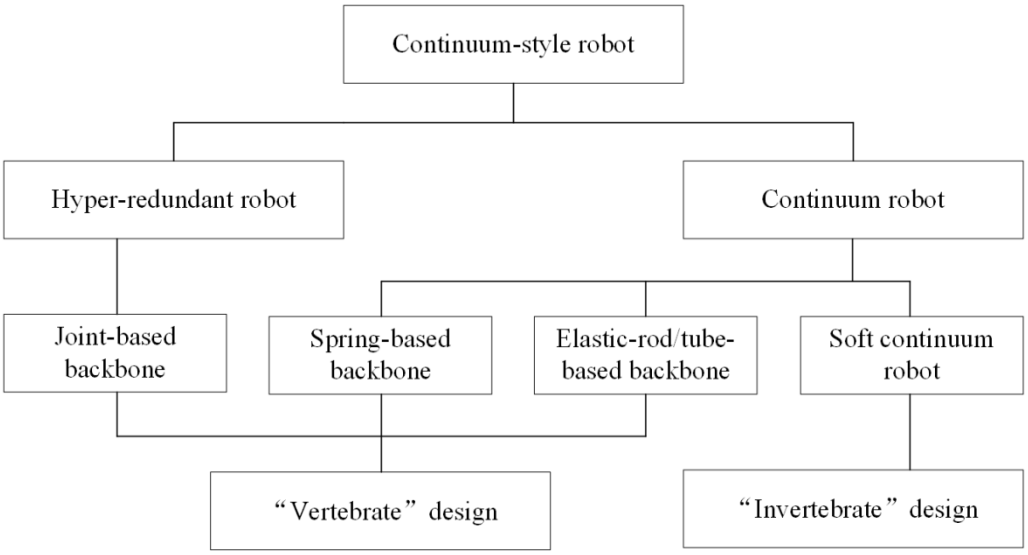


Figure 3-1 The classification of continuum-style robots according to their distinctive backbone architectures.

on the flexure joint and material properties. One main advantage of using this design is a large open lumen within the metal shell.

Another popular design of continuum robots utilises a laterally super-elastic, but longitudinally incompressible rod/tube as the backbone element. This continuous backbone design is commonly wire actuated (Gravagne, Rahn, & Walker, 2003); however, there are exceptions from this rule, in which case the wires are replaced by a secondary backbone and bending is achieved by means of a push-pull mode (Simaan, 2005). A distinctive feature of using an elastic central backbone is design simplicity. On top of this, both control and modelling will be simplified. Some commonly used medical devices share this feature as well, in (Camarillo, Milne, Carlson, Zinn, & Salisbury, 2008) for example a catheter is introduced, which can be seen as an elastic tube steered by tendons. Furthermore, a multi-section design emerged, producing S-shape and more complex configurations. The following is a special case of a continuum manipulator, known as active cannula or concentric tube, proposed by Webster et al. (Webster, Joseph, & Cowan, 2009), (Burgner, et al., 2014) and Dupont et al. (Dupont, Lock, Itkowitz, & Butler, 2010). The main idea is to utilise pre-curved super-elastic tubes to achieve various backbone shapes. The bending of a continuum manipulator can be formulated by beam-mechanics-based models. More recently, the ideas of layer jamming (Kim, Cheng, Kim, & Iagnemma, 2013) and granular jamming (Jiang, Xynogalas, Dasgupta, Althoefer, & Nanayakkara, 2012) have been exploited to achieve tunable stiffness of the continuum robot structure. They differ from previous designs being composed of multiple jamming

elements. The structure stiffness is changed by applying vacuum pressure to control the friction between granular media.

Except for the aforementioned types of continuum manipulators that employ different modalities of a backbone structure, there is also the “invertebrate” design. It is often termed “bio-inspired soft continuum robot”. In this case, continuum manipulators are integrated with artificial muscles but do not possess a backbone or spine (Pritts & Rahn, 2004), (McMahan, et al., 2006). The designs directly resemble animals or animal appendages, for example octopus tentacles; different actuation mechanisms are used. Most often, continuum structures are pneumatically actuated – an approach closest to a muscular hydrostat (Kier & Smith, 1985). That not only enables the robot to achieve elongation/shortening and bending motions, but also provides the capabilities of stiffening and force generation. Recent advances in both soft actuators and sensing techniques have led to an increased interest in the use of soft structures in continuum manipulators, however, it is still an inchoate field and a number of fundamental scientific issues need to be addressed. In particular, the fabrication and control of soft bodies would be challenging. It is expected that this kind of continuum manipulators embodies new robotic concepts and leads the design of mechanisms to the essence of biological systems.

So far, these fundamental, related, and broader continuum-style manipulator designs were reviewed. Nevertheless, the diversity of design strategies is not limited to the afore-mentioned structures. We seek to put our work in the context of other continuum manipulator designs and discuss its advantages and negative aspects. It was recognized that no single design is perfect.

3.1.1 Advantages of the proposed design with planar springs

One primary advantage of using compliant planar springs is due to their linear output motion. This will be experimentally verified later with 3D-printed plastic modules and their assembled continuum manipulator. In Chapter 4, we will systematically generalize the analytical method to study the compliance characteristics of the planar spring and provide the unified compliance matrix to represent the force-deflection relationships, thus allowing us to accurately predict the robot’s motion.

Another advantage of this continuum manipulator is due to the serial connection of the conjoined layers, thus allowing linear predictable bending even for large bends, although the linear-motion approximation of each individual layer only holds under the condition of small deflections. The proposed structure behaves like a helical spring, but its contraction and bending motion are decoupled, thus virtually eliminating any uncontrolled compression when generating bending deflections. Additionally,

the structure is longitudinally compliant – a desirable feature in robotics, which improves safe interactions. On the other hand, the compressible manipulator length extends the achievable workspace and enhances the dexterity tuning manipulator's tip orientations.

However, due to the complex design of its structure, the fabrication is a complex process. Currently, the author builds the manipulator prototype using 3D printing technique, but in practical applications that require to achieve some degree of stiffness and robustness, a construction from metal is necessary – to construct such a multilayer structure form metal is challenging. Besides, due to its continuum backbone being constructed by multiple layers of planar springs, there is no open lumen inside the continuum manipulator compared to a flexible tube, and this in some cases will be another limitation.

A 3D-printed prototype of the proposed continuum manipulator is presented in Figure 3-2. The continuum structure is actuated in the three-tendon-driven manner, and the motor package is located on the proximal end of the continuum manipulator.

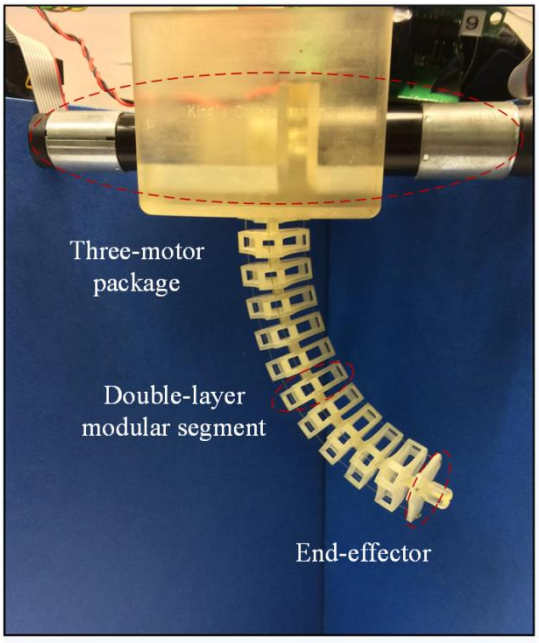


Figure 3-2 A 3D-printed prototype of the proposed continuum manipulator concept with multilayer deformable planar springs in series and articulated by tendons which are actuated by motors in the periphery.

3.2 Conceptual design

3.2.1 Segment design

Figure 3-3(a) depicts a top view of the compliant planar module. It is usually called “planar spring”. The planar spring is a compact spring and it can be fabricated from a piece of planar material. The three structural elements include central platform, suspending leg, and outer base. In the development, Parise and Howell et al. first constructed this type of designs and identified different configurations (Parise, Howell, & Magleby, 2001), (USA Patent No. WO2001081785 A1, US20040021123 A1, US6983924 B2, 2000), (Howell L. L., 2001) and considered them as a unique type of planar springs. We further put forward that this planar spring not only undergoes an out-of-plane motion along an axis orthogonal to the parent plane, but also has the potential for angular deflection. In Figure 3-3(a), the radial-leg design is presented in detail: three legs (120° apart) radially extend away from the central platform and are anchored to the outer base; each leg has two flexible segments shaped like a “U” (U-shape design); the intermediate platform is considered infinitely stiff. In the current prototype design the circular outer contour has a 29mm diameter and the length of each leg is 8mm. The thickness of the flexible beam elements is 1mm; the width 1.2mm and both can be varied to change the beam compliance. Part of the base is cut in order to reduce the mass. Three tendon channels with a 0.8mm diameter are reserved for guiding tendons through each module layer. The tendons are positioned on the far edge of the base and along the extension line of the leg. Optimization of the design is needed regarding different practical specifications and fabrication materials. Note that all these dimensions used here and in the following are only intended for a 3D prototype with the plastic materials, and as for the further fabrication with metal materials, for example using super-elastic Nickel-Titanium (NiTi) alloy, the design parameters has to be reconsidered depending on the properties of metal materials.

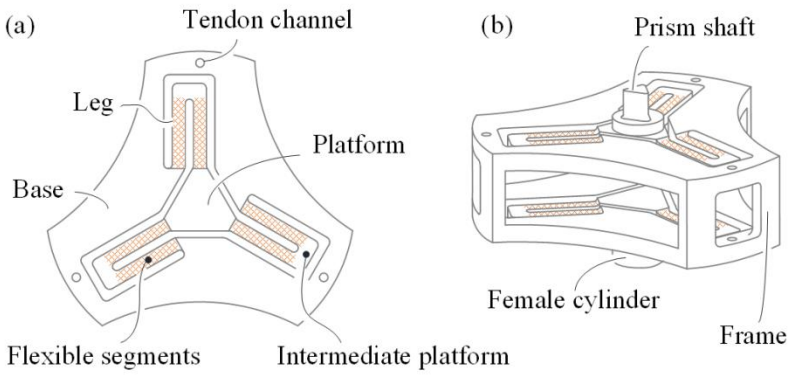


Figure 3-3 Schematic of double-layer modular segment. (a) Top view; (b) Side view. The pattern-filled parts represent the flexible segments.

Due to the elastic flexing of the slender leg beams, this compliant planar spring possesses motions to raise and lower the platform relative to the fixed base and to allow the platform to freely perform titling motions around the centre. A three-legged design is chosen because having three legs is the minimum odd number leg count to allow reducing the rotational tendencies of each leg and increasing the stability of the platform (Parise, Howell, & Magleby, 2001)

Figure 3-3(b) depicts the modular segment design for our continuum manipulator. It integrates two layers of compliant planar springs opposing each other; a prism-like shaft and a mating female cylinder are respectively fixed on each platform at the top and the bottom. The polygonal cross-section design of the axial coupling resists relative rotation between the two segments whilst enabling torque transmission. They are fitted precisely to connect from segment to segment. This design simplifies the assembly of the current 3D prototyped modules. Except for the flexible segments and the two platforms with their "vertebrae", any other part of the segment is a part of the frame, see Figure 3-3(b), which is idealized to be a rigid body. If fixing the bottom cylinder rigidly to the ground and applying a load to the prism shaft, the relative displacement and/or rotation of the two platforms would be double when compared to that of one layer for the same load. The gap between the two layers currently is 5mm, providing enough space to keep the deformed legs or two platforms of the top and bottom layers, respectively, from colliding during a bending motion. The segmented modular design allows the length of the continuum manipulator to cope with various intended, bending scenarios.

3.2.2 Continuum robot assembly

Our current continuum manipulator prototype consists of 10 modular segments. Figure 3-4 shows partial views of the assembly. For detailed CAD drawings of the whole continuum robot with the actuation system, please refer to Appendix A. Including a distal plate and a bottom support, the total length is 143mm. The distance between the lower layer of one segment and the upper layer of the subsequent segment is 5mm and, in our prototype, the same gap of 5mm is chosen as distance between the upper and lower layers of each segment. Three tendons are routed along the aligned segments through tendon channels and secured to the distal plate, which is connected to upper layer platform of the last segment; this approach leads to a tendon-driven under-actuated design. By pulling the tendons, the load will be transmitted from the distal top platform to the proximal bottom support, thus generating compression and steering motions. Moreover, depending on the intended operations, the design could be extended by incorporating additional groups of tendons to increase the mobility and functionality. The set of tendons are secured to some selected pillar and produce torques to the section down to the tendon termination point.

Other continuum robots utilizing local spring elements along the backbone have appeared and been presented in (Yang, Jason, & Abdel-Malek, 2006), (Mehling, Diftler, Chu, & Valvo, 2006), (Watanabe, Kanou, Kobayashi, & Fujie, 2011), (Hannan & Walker, 2003), however, these spring elements are the basic helical springs. The design features of the proposed continuum robot in this paper are utilizing a type of planar springs and introducing the double-layer modules.

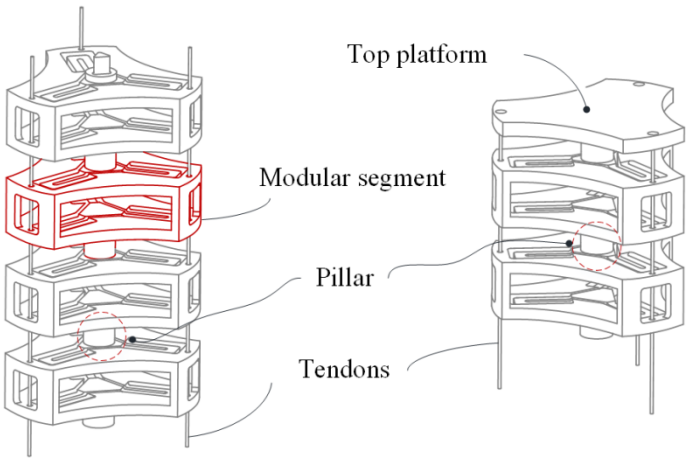


Figure 3-4 Partial views of continuum manipulator assembly.

3.3 Design variations

3.3.1 Design variations of planar springs

In Figure 3-5, three U-shape legs are distributed around the central point of the structure at 120 ° intervals and they enable the central platform to move or rotate relative to the outer base but constrain any rotation about the vertical axis. Thus, the central platform has one degree of freedom (DOF) of raising and lowering along the vertical axis of the spring plane and two DOFs to allow the platform to freely perform tilting motions around the centre about two orthogonal axes. In total, the central platform has three DOFs for actuation if the outer base is fixed, and conversely, if the central platform is fixed, the outer base possesses equivalent three DOFs. The central platform and outer base of the planar spring are considered infinitely stiff, whilst the rest U-shape legs are flexible (the intermediate segment bridging between two beams is considered infinitely stiff as well). There can be less or more legs for suspending the central platform; however, as discussed before, the three-leg design allows increasing the stability of the platform while keeping the number of legs to a minimum. Besides, the U-shape leg can be substituted with the N-shape leg which now consists of three fixed-guided beam flexures; the E-shape leg is also an alternative approach to suspend the central platform with outer two flexures connecting the outer base and middle flexure anchored to the platform (Parise, Howell, & Magleby, 2001), (USA Patent No. WO2001081785 A1, US20040021123 A1, US6983924 B2, 2000). However, from the perspective of both the structural simplification and stability, the U-shape leg is superior to other options, thus, only this U-shape leg configuration is considered for the current design.

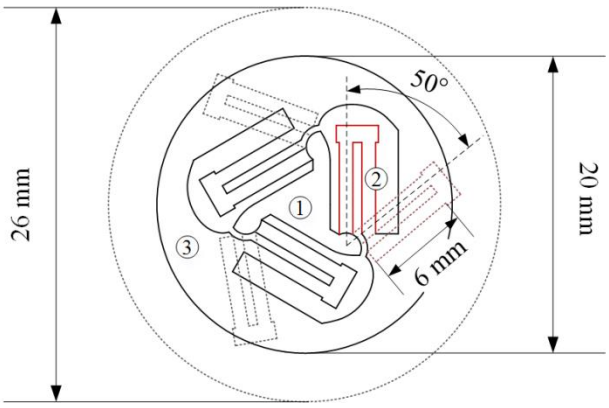


Figure 3-5 Structural design of the planar spring with varying angles of leg attachment: (1) central platform, (2) suspending leg (U-shape, in red colour), (3) outer base. The solid-line contour represents a standard side-leg design of the planar spring; the dotted-line contour illustrates a configuration with the legs swinging out 50˚ from the standard side-leg positions, which we call "titled-leg design". With the changing of the leg angles of attachment, the diameter of the outer base increases from 20 mm to 26 mm.

Figure 3-5 also depicts one way to create other design variations of the planar spring in the framework of U-shape three-leg designs. More illustrations of different configurations with the varying angle of leg arrangement are shown in Appendix B. It is feasible to attach the legs with various angles from the edge of this triangle platform. In this case of the Figure 3-5, the angles of attachment can range from 0˚ (the standard side-leg designs in (Parise, Howell, & Magleby, 2001)) to 120˚ (the standard radial-leg designs in (Parise, Howell, & Magleby, 2001)). Under this system we call these design variations with inclination angles of attachment titled-leg designs. Accordingly, the diameters of the outer base of titled-leg designs rise as their angles of attachment increase. The side-leg design reveals to be very compact. With the same leg length and the same platform size as configurations in Figure 3-5, the planar spring in side-leg configuration shows a smaller diameter. In the current design for prototyping with acrylonitrile butadiene styrene (ABS) plastic material, the length of all legs is configured as 6mm; the circular outer contour diameter of the side-leg design is 20mm; the titled-leg design is 26mm. Smaller size structures can possibly be made using other materials such as metal.

3.3.2 Design of a conic continuum manipulator

Figure 3-6 depicts a 3D-CAD drawing of a partial assembly of the integration of multiple variations of planar springs in series to form a conic continuum manipulator. The depicted configuration is in its

bending status with an angle 60 °. The flexibility of the continuum manipulator is produced by these flexible legs, which is evidenced by the equivalent stress rendering in the U-shape legs. The distance between any two adjacent layers of planar springs is fixed and constructed with rigid frames pictured in grey (with a 70% transparency) in Figure 3-6. The designs of the prism-like shaft and the mating female cylinder shown in Figure 3-3 are replaced with pillars in this drawing. The module segment is also divided into two separate planar spring layers and the rigid frame in order to assign different material properties, respectively. The pillars connect central platform to platform; the ring-shaped walls connect the outer base to base; both are rigid, arranged alternatively, and play a supporting role. With various angles of the leg arrangement, each layer of planar springs demonstrates a different bending behaviour – the angular displacement decreases in direction to the manipulator bottom. This change rule with respect to compliance analysis will be revealed in details in Chapter 4. This continuum manipulator can deflect in 3D space towards any directions and also has the abilities to contract and elongate depending on axial forces applied to it.

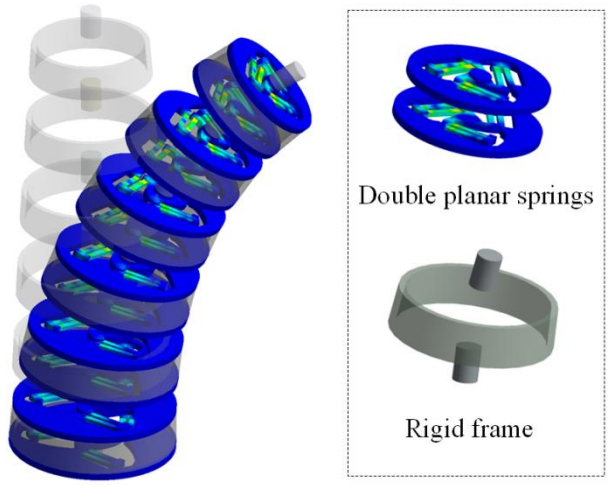


Figure 3-6 Part of a conic continuum manipulator assembly. It integrates multiple layers of planar springs that have various angles of leg attachment. The rigid frames are shown in a transparent mode in order to reveal all layers of planar springs. The bottom of the manipulator is fixed to the ground and the top is exerted with a moment; the resulting bending status is in its 60 °with the equivalent stress rendering. The double planar springs as illustrated separately are assigned with flexible materials, and their connecting frame shown below is rigid, including two pillars and a belt-shape wall.

3.4 Experimental tests

This section presents experimental results to verify the performance of the proposed continuum manipulator. The tests validate the cumulative large linear bending characteristic, decoupling property between bending and compression of this continuum manipulator, and its enlarged workspace.

3.4.1 Experimental setup

The double-layer modular segment is made of UV curable acrylic plastic material (VisiJet® EX200) and is 3D printed using a rapid prototyping machine (ProJet™ HD 3000, Resolution 328×328×606 DPI (*xyz*)). Other geometrical dimensions remain the same as before. Modular design here allows the overall length and maximum bending to be easily modified to suit the intended applications.

Figure 3-7 shows the experimental test platform and environment, where the double-layer module and the multi-segment assembly can be mounted to the holder at one side of a linear guide (KK40-2001, Hiwin). One non-stretchable transmission tendon was routed through the segments and one end terminates at the distal plate (as illustrated in Figure 3-4) and another end is fastened to a commercial force sensor (ATI® Nano 17 6-DOF force/torque sensor) that is used to record the reaction pull-force along the tendon. In order to eliminate the gravity effect, the segment/continuum manipulator was laid on the horizontal plane and only horizontal bending motion is performed via the tendon actuation. The force sensor was held on a moving block attached to the linear guide allowing only linear translation. The taut tendon is always perpendicular to the holding block and corresponds to a consistent moving direction of the slider. The linear guide was driven by a DC motor (Maxon Motor®) that was connected to a positioning controller. A LABVIEW® program was developed to control the linear guide with the desired speed and to record the reaction force data acquired by the force sensor.

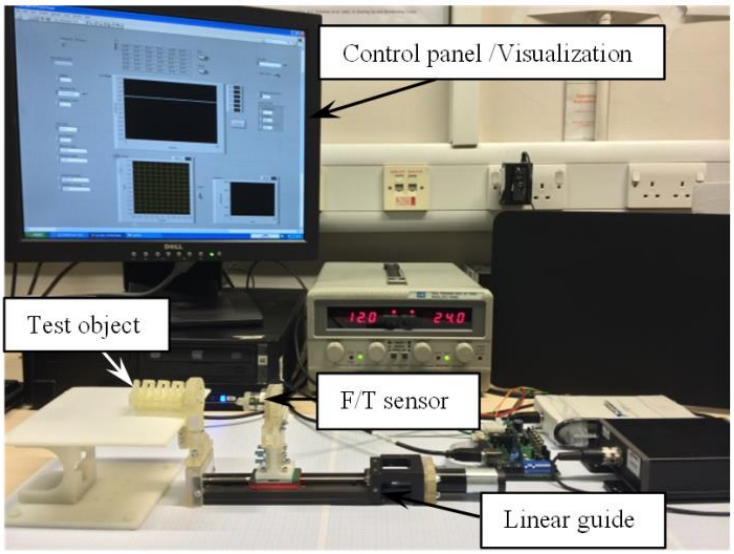


Figure 3-7 Experimental setup for modular segment test and quantitative analysis.

3.4.2 Modular segment tests

The experiments start with single modular segment tests and then three-segment assembly tests. Based on the experimental results, the linear and decoupled behaviours of this multilayer planar spring based design are experimentally investigated. Figure 3-8 shows several snapshots of the configurations of a double-layer module and a three-segment assembly when experiencing known loads applied via tendon(s). The other end of the string is tied to the Nano 17 force/torque sensor (sensitivity: 0.318 gram-Force) which is held on the slider and moves along the rail quasi-statically and slowly at a constant speed of 10mm/min, thus, forces are produced along the taut tendons.

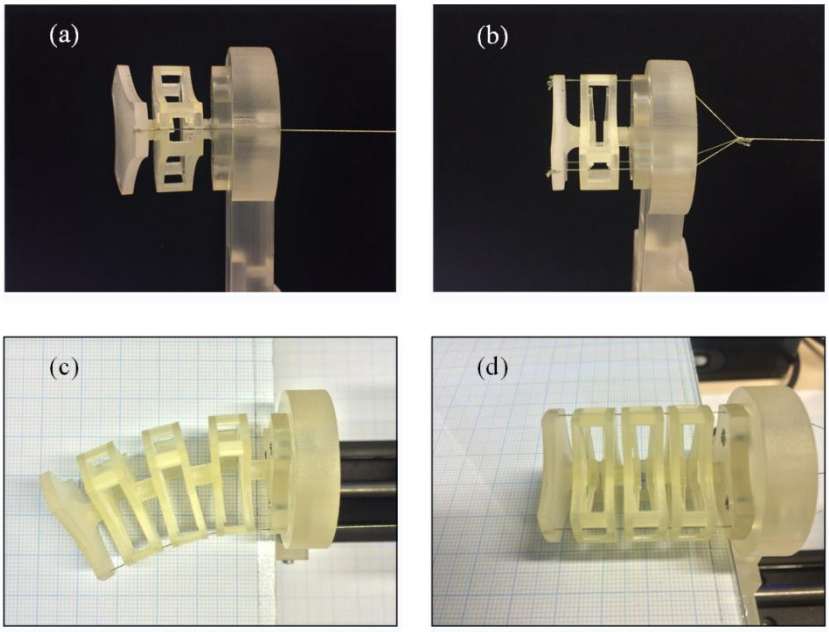


Figure 3-8 One modular segment and three-segment assembly , respectively, are mounted to the holder at one side of the linear guide. Forces are exerted via tendon(s): (a) one segment moment-rotation test; (b) one segment contraction test; (c) three-segment assembly moment-rotation test; (d) three-segment assembly contraction test.

3.4.2.1 Results of segment bending tests

Figure 3-9 illustrates the measured force-displacement curves for the moment loads acting on a double-layer modular segment and a three-segment assembly. A single tendon is utilised in this scenario. Both the *x*-and *y*-values are directly from the collected sensor readouts. The curves are closed loops, revealing the pulling and releasing tendon processes – hysteresis behaviour is exhibited. We suspect that the hysteresis is mainly due to material properties and is partly because of the changes in the direction of the friction force. The influence of material properties is a stress relaxation phenomenon where a decrease in stress is observed in response to the same amount of applied strain. In the end of both releasing tendon processes, the total amount of such plastic strain is depicted in magenta in Figure 3-9. The single segment relaxation indicated in Figure 3-9 is 0.29mm and it is 7.25% of the total strain measured during this test. The three-segment assembly relaxation is 0.90mm and it is 9% over the total strain. The increase of stress relaxation is due to the accumulation effect of the multiple segments.

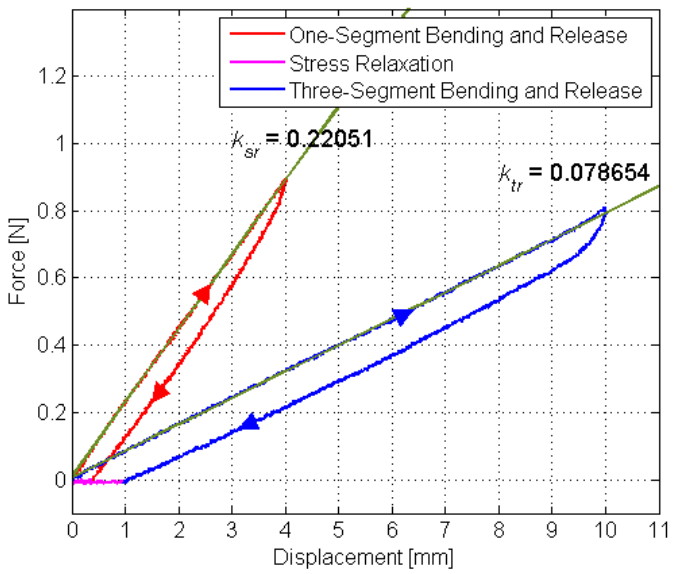


Figure 3-9 Plot of experimental results regarding the moment loads on one modular segment and one three-segment assembly. The ATI® Nano 17 F/T sensor is attached to the holder in a way to allow only one axis force readout along the tendon direction.

3.4.2.2 Discussions of segment bending tests

The forward path corresponding to the pulling tendon process in both one-segment and three-segment tests, as depicted in Figure 3-9, displays strong linearity. The reverse path corresponding to the releasing tendon process shows a smooth transition at the very beginning but then exhibits the linear feature. For both tests, the steady slopes of the forward and reverse paths are almost equal. A robust regression technique is utilised to formulate the forward path of each test; the derived slopes k_{sr} for single module test and k_{tr} for three-segment test are labelled in Figure 3-9, respectively. It is noted that there exists a multiple relationship between the two slopes ($k_{sr}/k_{tr} \approx 3$), and this matches the fact that the number of segments of the test object has tripled.

This empirical test validates the linearity of the stress-strain relationship. Since the properties of these 3D-printed parts are not completely consistent with the original printing materials, even parts made within the same printing batch exhibit different mechanical performance, thus the comparisons between the experimental and simulation results given in next chapter will not be further discussed.

We continue to study the bending configuration of the three-segment assembly in this test and examine its total length. It shows that the maximum coupled contraction is within 4% which is negligible and we can regard the length being invariant. Therefore, these experiments confirm the predicted decoupling between bending and contraction. This is an important aspect and the primary contribution of

our multilayer planar spring stacked concept, and, as such, very different from the way helical springs behave. Besides, we observed that each layer's bending angle or contraction distance was about equal and uniformly distributed across the total deformations in all the four tests, Figure 3-8. The maximum reachable bending angle depends on the material properties and segment design.

3.4.2.3 Results of segment contraction tests

Figure 3-10 shows the measured stress-strain curves for the forces exerted perpendicularly to a double-layer modular segment and a three-segment assembly. Motion snapshots are shown in Figure 3-8(b) and Figure 3-8(d). In this scenario three tendons are utilised and they are tied together before they are affixed to the force/torque sensor. The advantage of this is that the test procedure is simplified with only one sensor being needed; however, this approach amplifies the frictional forces between tendons and tendon channels. Since the frictional forces act against the direction of pulling tendons, the reaction forces recorded using the force sensor will be higher than the loads on the test objects. The reverse paths also include the effect of the frictional force but in the different direction, which reduces the tendon tension. In order to reduce the friction, applying the lubrication to tendons and Teflon coating and if possible reducing the number of actuated tendons are suggested.

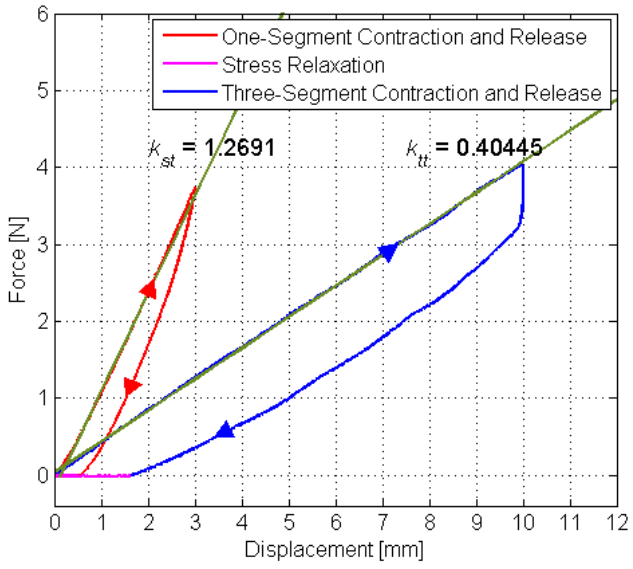


Figure 3-10 Plot of experimental results regarding the forces exerted on a segment and a three-segment assembly. The same procedure with the bending tests is followed. The tendon tension is given by the ATI® Nano 17 F/T sensor.

3.4.2.4 Discussions of segment contraction tests

The curves in Figure 3-10 exhibit the same linear trend as those in Figure 3-9, thus, we come to same conclusions as the bending tests in terms of linearity, multiple relation, and hysteresis behaviour. The contraction rate here is limited to 80% of the original length and the total relaxation percentage over the total tested contraction is 18.2 % for one segment test and 15.9% for the three-segment assembly test. This is much larger than the bending deformation.

Regarding the exhibited hysteresis, we suspect that hysteresis is caused by the stress relaxation of plastic materials. It is noted that using other more homogeneous and low-hysteresis materials, such as aluminium alloy and super-elastic NiTi, to fabricate the compliant planar springs, can reduce the hysteresis of the modular segment as well as of the assembled continuum body. However, in order to control such a nonlinear system in practise, future work on the analysis of the system's dynamics may help further understand and control hysteresis.

3.4.3 Continuum manipulator prototype experiments

A 3D-printed prototype of the multilayer structured continuum manipulator was tested; test procedures and results are described in this subsection. The length of this 10-segment assembled continuum structure is 143mm and its diameter is 29mm. Three tendons at the periphery are independently driven by three DC motors (Maxon Motor®) with pulleys. The 128:1 reduction gearhead employed allows tendon actuation with a high rotational resolution. The previous three-segment assembly can be used as a flexible robotic tip/wrist, and this continuum manipulator due to its much larger bending and contraction ranges can be used in different scenarios with different end-effectors, such as a detachable gripper, sensor module or cutting tools.

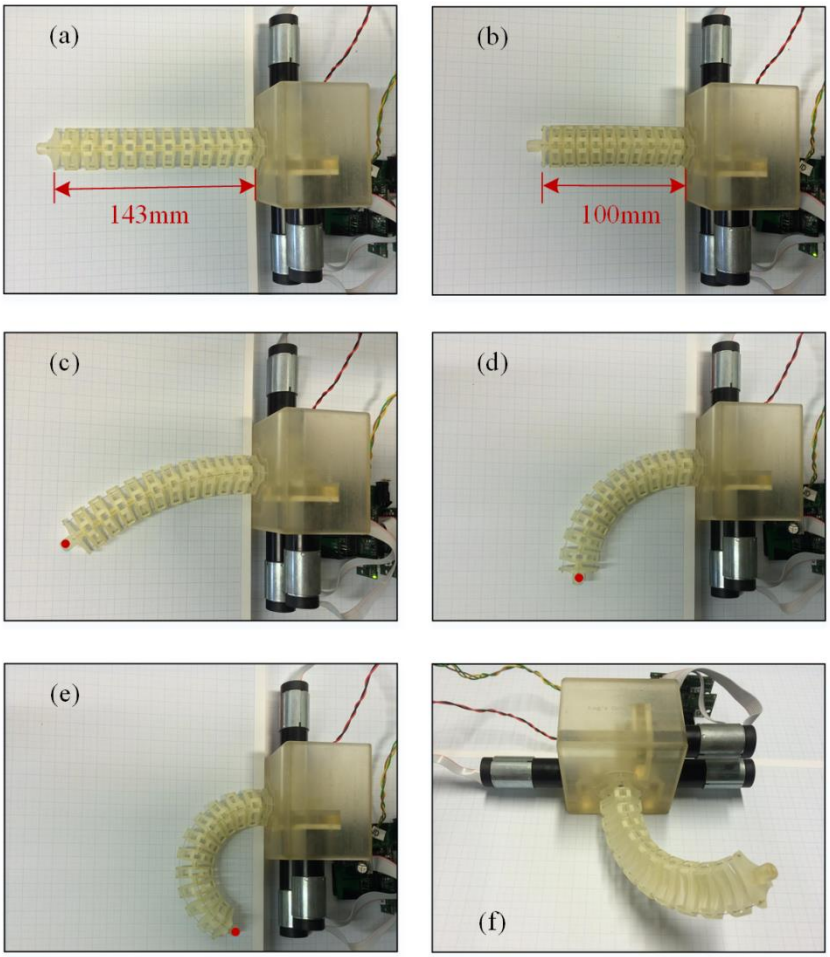


Figure 3-11 The proposed continuum manipulator performing bending and contraction deformations. The tip is a mock end-effector. (a) Before and (b) after longitudinal contraction; (c) – (e) 2D bending motion; (f) 3D bending motion.

Figure 3-11 shows several snapshots of the manipulator's bending and contraction configurations. Figure 3-11(a) and Figure 3-11(b) present a comparison of before and after the longitudinal contraction, which indicates the longitudinal compliance and, to some extent, ensures safety when interacting with the environment. Due to the limited compliance of the fabrication material, the prototype only serves as a preliminary setup for the investigation of the performance of contraction and bending motions. The contraction ratio of the currently fabricated prototype is 0.7, which is determined by material properties, dimensions of the flexible leg segments, and the structure constraints between two adjacent planar springs. On one hand, the longitudinal compliance characteristic is an advantage compared to longitudinally incompressible rod/tube based continuum manipulators. On the other hand, compared to the helical-spring-backbone-based continuum manipulator it has the capability of maintaining better structural rigidity for the whole continuum body.

Figure 3-11(c) – (f) illustrates the bending motion of the manipulator. The continuum manipulator is actuated by three tendons simultaneously and this enables the manipulator to deflect in 3D space (see Figure 3-11(f)). In order to determine the workspace of the manipulator tip, the 2D tip positions in the horizontal plane are marked on coordinate paper. The position is later normalized as shown in Figure 3-12. Theoretically, the corresponding 3D workspace is the area of a 2D workspace swept about its central vertical axis. This results in a large number of possible bending configurations. Several of these configurations were selected for the experiments. The central length of the bending manipulator was then measured for each configuration. The results are listed in Table 3-1. It is noted that the length of the manipulator is almost invariant however with the bending angle increasing to values higher than 30 degrees certain levels of contraction are exhibited. The maximum contraction rate is less than 6% and given the fact that the material is non-homogenous, we can still consider the bending motion to be decoupled from the contraction. The bending and contraction decoupling characteristic makes the design simple to control, while for the helical-spring-backbone based designs or tendon-driven pneumatic backbone based designs, the control effort leading to a bending actuation is partially lost in compression (Walker, 2013). From Figure 3-12, we can see that the red trajectory indicates the tip positions of a spring backbone continuum manipulator when exerted with a bending force.

In Figure 3-12, the outermost curve pictured in green indicates the tip trajectory of the continuum manipulator without control efforts to generate compressions. When contracting the manipulator, the bending deflection results in the tip trajectory shrinking to the inner curve layers. Due to the structure constraints, the maximum sweep range of a contracted continuum manipulator is narrowed. It is obvious that the inner curve is always slightly shorter than its adjacent outer curve. All the reachable tip positions form a crescent workspace, which verifies the enlarged workspace compared to a constant length manipulator, for example, a longitudinally incompressible rod/tube based continuum structure.

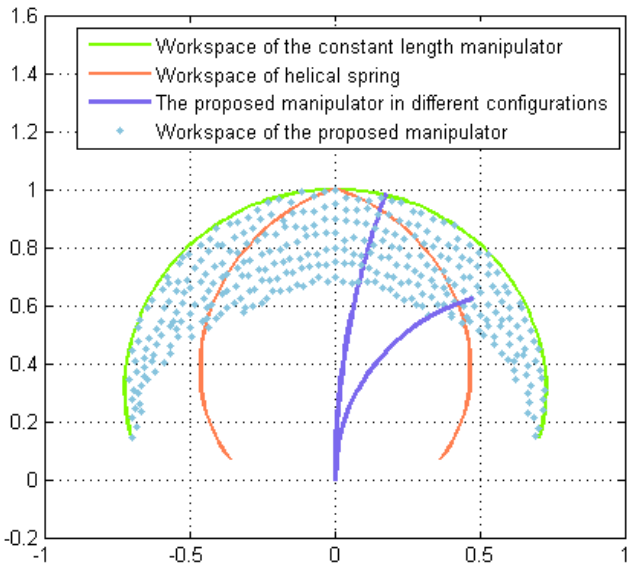


Figure 3-12 Normalized workspace of the proposed continuum manipulator compared with constant-length and helical spring backbone based continuum manipulators.

Table 3-1 Experimental results for different bending configurations.

Bending angle	***Manipulator length***	***Contraction***
0 °	143mm	0
10 °	143mm	0
20 °	143mm	0
30 °	142mm	0.7%
60 °	141mm	1.4%
90 °	140mm	2.1%
120 °	138mm	3.5%
160 °	135mm	5.6%

3.5 Discussions

This chapter presents the design of a continuum manipulator with multiple layers of compliant planar modules linked in series. Firstly, we reviewed frequently applied continuum manipulator constructions to date based on the distinctive backbone architecture. Through our study, we found that our structure has advantages over other existing traditional continuum manipulators. These advantages are longitudinal compliance, large linear displacement motions, effectively decoupled contraction and bending motions, as well as an enlarged workspace. We built and tested a prototype and proceeded with a series of experimental studies. The results verified the claimed characteristics of the manipulator. In the next

chapter, we will derive the compliance matrix of the planar spring and conduct FEA simulations to further confirm the predicted behaviour.

We also discussed some basic design variations of the planar springs. Future work will include a broader investigation on different design variations of this type of planar springs as the compliance characteristics of the planar spring are determined by its geometric parameters. We have already noted that changing the U-shape legs' position would enable the design to be more compact whilst largely increasing the compliance of the planar spring. Further, we may integrate different designs into one continuum manipulator and apply several more groups of tendons to actitation different sections of the continuum manipulator, such that the bending configuration can demonstrate various curvatures as well as various compliances along the continuum manipulator.

Moreover, we will fabricate the continuum manipulator with other materials such as metal, thus smaller size will be made possible and the hysteresis behaviour is expected to be reduced. Potential applications will be studied with suitable end-effectors attached, such as a detachable gripper for medical applications. Besides, in this tendon-driven design, the friction leads to undesirable effects, thus possible methods to reduce or compensate the friction effects will be studied.

Chapter 4 Modelling of Continuum Manipulators

In Chapter 2, the mathematical frameworks to describe and analyse the continuum manipulator have been presented in a rigorous manner. In this chapter, we will utilise these mathematical tools to study the compliance characteristics, kinematics, and dynamics of the continuum manipulator. Firstly, upon the experimental studies of the planar-spring-based design of the continuum manipulator in the last chapter, an analytical method is provided to study the compliance characteristics of the planar spring and derive the compliance matrix to represent the force-deflection relationships, allowing an accurate motion prediction. Further the compliance matrix of planar springs reveals the six-dimensional compliance characteristics being determined by the layout, and the material and geometric parameters. Differences of the compliance properties with respect to design variations of planar springs are discussed. FEA simulations are conducted to validate this analytical model.

Secondly, we investigate the continuum manipulator kinematics and elucidate the two-step kinematic mapping related to continuum manipulators which is different from conventional rigid-link manipulators. The kinematic model of our applied tendon-driven continuum manipulators is derived according to the constant-curvature approximation. This modelling methodology permits closed-form kinematics and also facilities the derivation of differential kinematics and real-time control. Besides, we also present an overview of the relations between continuum manipulator Jacobians and statics after deriving the compliance matrix of the full robot. Moreover, another popular case of a single-tendon-driven continuum manipulator with a rotary motor for body rotation is discussed, and its kinematics and Jacobians are presented.

In the end, the Lagrangian dynamics of the continuum manipulator is investigated based on the curve geometry presented under the Frenet-Serret frame that is established in Chapter 2. Thus, we establish a comprehensive mathematical structure for studying the continuum manipulator.

4.1 Six-dimensional compliance matrix derivation

From the perspective of mechanical design, this planar spring is a type of hybrid flexure mechanism (Parise, Howell, & Magleby, 2001). Each flexible segment in each leg can be treated as a beam flexure. Each leg is a folded serial chain of two fixed-guided beam flexures. This planar spring is formed by connecting the central platform to the outer base through three legs in parallel. Thus, we can stepwise derive the compliance matrix for the entire module with a bottom-up approach. Stiffness and compliance analysis for the general robots has also been represented in other different ways before (Roberts, 1999), (Roberts, 2000).

4.1.1 Compliance matrix of an elastic beam

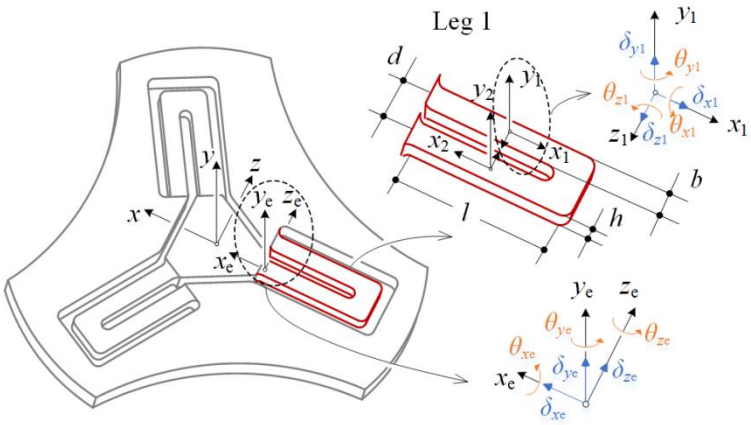


Figure 4-1 Illustration of local and global coordinate frames on the planar spring plane.

Figure 4-1 depicts the establishment of the local and global coordinate frames. The linear elasticity is considered throughout the derivation. In the framework of screw theory (Dai, 2012), a small deformation of a beam is defined as a twist deflection, which in axis coordinates can be denoted by

$$\mathbf{S} = \begin{bmatrix} \delta_x & \delta_y & \delta_z & \theta_x & \theta_y & \theta_z \end{bmatrix}^T \tag{4-1}$$

where the first three elements represents the three translational deflections along their corresponding axes, whilst the last three elements reveal the corresponding rotational deflections. A twist deflection **S** is an element of the Lie algebra *se*(3) of Lie group *SE*(3).

In-line with (4-1), the loading force is considered as a general wrench in ray coordinates

$$\mathbf{W} = \begin{bmatrix} f_x & f_y & f_z & m_x & m_y & m_z \end{bmatrix}^T \tag{4-2}$$

in which the location of the axis of the wrench is given by the primary part $\boldsymbol{f} = \begin{bmatrix} f_x, f_y, f_z \end{bmatrix}^T$, whilst the secondary part $\boldsymbol{m} = \begin{bmatrix} m_x, m_y, m_z \end{bmatrix}^T$ is the vector attached with the force intensity, representing the direction of the axis of the wrench. A wrench is an element of the dual Lie algebra *se**(3).

Consider one beam of the leg (pictured in red in Figure 4-1), a local coordinate frame $\{x_1y_1z_1\}$ generally can be established at the centroid of the beam. With the coordinates of both the twist deflection and the wrench written in the same frame $\{x_1y_1z_1\}$, then the compliance matrix of this beam can be derived (Selig & Ding, 2001) and expressed as

$$\mathbf{C}_1 = diag\left[\frac{l}{EA} \quad \frac{l^3}{12EI_z} \quad \frac{l^3}{12EI_y} \quad \frac{l}{GJ} \quad \frac{l}{EI_y} \quad \frac{l}{EI_z} \right] \tag{4-3}$$

where the primary part represents linear compliance and the secondary part the torsional compliance. The beam has a length l and a rectangular cross-section with the width b and the thickness h ($b>h$). The area of the cross-section is represented by A which is equal to bh, E denotes the elastic module of the material, and G denotes the shear module of the material with $G=E/(2(1+v))$ and v being Poisson's ratio. The moments of inertia of the beam at the cross-section with respect to axis y and axis z are $I_y=b^3h/12$ and $I_z=bh^3/12$, respectively, and the torsional moment of inertia is described by J.

Equivalent results are also produced in (Selig & Ding, 2001), (Su, Shi, & Yu, 2012) and there exists remarkable similarity, however, due to coordinate frame choices, they are diverse in form. The compliance of an individual link or a whole mechanism system is their intrinsic property, but the expression of the compliance matrix may vary depending on the coordinate frame choice.

4.1.2 Compliance matrix of the planar spring

For (4-1), (4-2), (4-3), we have the relations between a twist deflection and a loading wrench as summarized below

$$\mathbf{S} = \mathbf{C}_1\mathbf{W}\ ;\ \mathbf{W} = \mathbf{K}_1\mathbf{S}\ ;\ \mathbf{C}_1 = \mathbf{K}_1^{-1} \tag{4-4}$$

where $\mathbf{K}_1$ is the stiffness matrix in the local coordinate frame $\{x_1y_1z_1\}$.

The second beam of the U-shaped leg is an identical flexible segment to the first one, thus the compliance matrix is the same but expressed in its own local coordinate frame $\{x_2y_2z_2\}$ as shown in Figure 4-1. Two beams in the leg are connected by an intermediate platform, but it is modelled as a fixed pin joint with its compliance ignored, when we consider the force-deflection relationship of the leg (Parise, Howell, & Magleby, 2001). At the connecting edge between the leg and the platform, we established the leg global coordinate frame $\{x_ey_ez_e\}$. To shift the local coordinate frame of each beam into the global coordinate frame $\{x_ey_ez_e\}$, an adjoint action of Lie group $SE(3)$ on its Lie algebra is introduced through a 6×6 matrix representation (Dai, 2012)

$$\mathrm{Ad}_g = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{DR} & \mathbf{R} \end{bmatrix} \tag{4-5}$$

where $\mathbf{R}$ is a 3×3 rotation matrix representing the orientation of frame $\{x_ey_ez_e\}$, relative to frame $\{x_1y_1z_1\}$; in this case, $\mathbf{R} = \mathbf{R}_y(\pi)$. $\mathbf{D}$ is a skew-symmetric matrix spanned by the position vector $\boldsymbol{d}$ of the origin of $\{x_1y_1z_1\}$ from the origin of frame $\{x_ey_ez_e\}$; in this case, $\boldsymbol{d} = \left[-l/2, 0, d\right]^T$.

Then, the coordinates of a twist deflection and a wrench in the coordinate frame $\{x_ey_ez_e\}$ are calculated as (Selig & Ding, 2001), (Su, Shi, & Yu, 2012)

$$\mathbf{S}' = \mathrm{Ad}_g^{-T}\mathbf{S} \quad ; \quad \mathbf{W}' = \mathrm{Ad}_g\mathbf{W} \tag{4-6}$$

To obtain the compliance matrix $\mathbf{C}_1'$ in the new coordinate frame, we deduct it as follows based on (4-4):

$$\mathbf{S}' = \mathrm{Ad}_g^{-T}\mathbf{S} = \mathrm{Ad}_g^{-T}(\mathbf{C}_1\mathbf{W}) = \mathrm{Ad}_g^{-T}\mathbf{C}_1\mathrm{Ad}_g^{-1}\mathbf{W}' \tag{4-7}$$

Thus, we find that the compliance matrix will be transformed to the new coordinate frame according to the relation

$$\mathbf{C}_1' = \mathrm{Ad}_g^{-T}\mathbf{C}_1\mathrm{Ad}_g^{-1} \tag{4-8}$$

Similarly, we can derive the stiffness matrix in the new coordinate frame $\{x_ey_ez_e\}$ as

$$\mathbf{K}_1' = \mathrm{Ad}_g\mathbf{K}_1\mathrm{Ad}_g^{T} \tag{4-9}$$

Here, the inverse and the inverse transpose of such adjoint transformation matrix are given, respectively, by

$$\mathrm{Ad}_g^{-1} = \begin{bmatrix} \mathbf{R}^T & \mathbf{0} \\ -\mathbf{R}^T\mathbf{D} & \mathbf{R}^T \end{bmatrix} \quad ; \quad \mathrm{Ad}_g^{-T} = \begin{bmatrix} \mathbf{R} & \mathbf{DR} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \tag{4-10}$$

A mathematical calculation for beam 1 is exemplified below. First, the rotation matrix **R** and the skew-symmetric matrix **D** for coordinate translation are written as

$$\mathbf{R} = \mathbf{R}_y(\pi) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{4-11}$$

$$\mathbf{D} = \hat{\boldsymbol{d}} = \begin{bmatrix} 0 & -d_z & d_y \\ d_z & 0 & d_x \\ -d_y & -d_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & -d & 0 \\ d & 0 & -\frac{l}{2} \\ 0 & \frac{l}{2} & 0 \end{bmatrix} \tag{4-12}$$

Then, substituting (4-11) and (4-12) into (4-5) yields the adjoint transformation of beam 1 as

$$\mathrm{Ad}_g = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -d & 0 & -1 & 0 & 0 \\ -d & 0 & -l/2 & 0 & 1 & 0 \\ 0 & -l/2 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{4-13}$$

Similarly, the adjoint transformation of beam 2 can be calculated via the same procedure, but notice that coordinate transformations are different.

If all deformations are presented into the same coordinate frame $\{x_e y_e z_e\}$, then the overall compliance matrix of the leg as a serial flexure chain is (Su, Shi, & Yu, 2012)

$$\mathbf{C}_{l1} = \sum_{i=1}^{2} \left(\mathrm{Ad}_g\right)_i^{-T} \mathbf{C}_i \left(\mathrm{Ad}_g\right)_i^{-1} = \begin{bmatrix} \frac{2l}{EA} + \frac{d^2 l}{EI_y} & 0 & \frac{d^2 l}{2EI_y} & 0 & -\frac{dl}{EI_y} & 0 \\ 0 & \frac{2l^3}{3EI_z} + \frac{d^2 l}{GJ} & 0 & \frac{dl}{GJ} & 0 & \frac{l^2}{EI_z} \\ \frac{dl^2}{2EI_y} & 0 & \frac{2l^3}{3EI_y} & 0 & -\frac{l^2}{EI_y} & 0 \\ 0 & \frac{dl}{GJ} & 0 & \frac{2l}{GJ} & 0 & 0 \\ -\frac{dl}{EI_y} & 0 & -\frac{l^2}{EI_y} & 0 & \frac{2l}{EI_y} & 0 \\ 0 & \frac{l^2}{EI_z} & 0 & 0 & 0 & \frac{2l}{EI_z} \end{bmatrix} \tag{4-14}$$

Given a compliance matrix of one leg, its corresponding stiffness matrix $\mathbf{K} = \mathbf{C}^{-1}$ is first calculated. It is noted that all twist deflections and wrenches here must be transformed into the same coordinate frame, and correspondingly, the stiffness matrix of each leg will be expressed in such a global coordinate frame. We

establish the module global coordinate frame {*xyz*} in the centre of the triangular platform, see Figure 4-1. The radius of the plate is labelled by parameter *r*. The coordinate transformation operation from the connecting edge between the leg and the platform, i.e. the edge of the disc to the centre of the disc follows the aforementioned relationship (see (4-9)). Further considering that the overall layer's stiffness is isotropic (Dai, 2014), it gives the unified form as

$$\mathbf{K}_0 = \sum_{i=0}^{2} \mathbf{N}^i \mathbf{K}_1' \left(\mathbf{N}^T\right)^i \tag{4-15}$$

where $\mathbf{K}_1'$ is the stiffness matrix of leg 1 in the global coordinate frame {*xyz*}; it is derived by the relation $\mathbf{K}_1' = \mathbf{T}\mathbf{K}_1\mathbf{T}^T$ based on (4-9), which indicates a coordinate transformation from the leg global coordinate frame at the connecting edge to the module global coordinate frame of the platform centre. In this case, **T** only possesses the translation action along the *x* axis. **N** describes the rotation action based on the fact that three legs are symmetrically connected to the platform with an angle of 120 °.

Finally, the compliance matrix of the overall planar spring as a type of hybrid flexure mechanisms is computed by inverting the stiffness matrix $\mathbf{K}_0$,

$$\mathbf{C}_0 = \mathbf{K}_0^{-1} = diag\begin{bmatrix} c_{11} & c_{22} & c_{33} & c_{44} & c_{55} & c_{66} \end{bmatrix} \tag{4-16}$$

Here, the nonzero compliance elements are denoted by the variables with two subscripts:

$$c_{11} = c_{33} = \frac{l^3\left(Ad^2 + 4I_y\right)}{3EI_y\left(3Ad^2 + Al^2 + 12I_y\right)}$$

$$c_{22} = \frac{l\left(3EI_z d^3 + GJl^2\right)}{18EI_z GJ}$$

$$c_{44} = c_{66} = \frac{4}{3}\cdot\frac{l\left(EI_z 3d^3 + GJl^2\right)}{EI_z GJ\left(6d^2 + 4l^2 + 12r^2 - 6\sqrt{3}lr + 6dr\right) + \left(EI_z\right)^2 3d^2 + \left(GJ\right)^2 l^2}$$

$$c_{55} = \frac{2}{3}\cdot\frac{l^3\left(Ad^2 + 4I_y\right)}{EI_y\left(A\left(5d^2l^2 + 9d^2r^2 - 6\sqrt{3}d^2lr + 2dl^2r\right) + I_y\left(16l^2 + 36r^2 - 24\sqrt{3}lr\right)\right)}$$

The compliance elements are all determined by both material parameters and geometric parameters of the mechanical design of the compliant planar spring structure.

4.1.3 Compliance analysis and numerical example

The compliance matrix in (4-16) is symmetric positive definite (SPD), and the diagonal entries represent the translational and rotational compliance in/about all directions, respectively. Besides, all diagonal compliance elements of $\mathbf{C}_0$ can factor out a factor that coincides with the corresponding elements of beam's compliance matrix in (4-3). By observing compliance elements of $\mathbf{C}_0$, we notice that the *x*-*z* planar motion (x, z, θ_y) is decoupled from out-of-plane forcing and vice-versa. On the other hand, the entries outside the main diagonal are all zero, revealing that the out-of-plane rotation and translational motion of the platform are decoupled. This further verifies that the contraction effort and bending motion of the multi-layer structured continuum manipulator will be theoretically independent to each other.

Referring to Figure 4-1, the current global coordinate frame {*xyz*} can be rotated about its *y*-axis and therefore the action in (4-15) needs to include a rotation matrix $\mathbf{R}_y(\varphi)$. Nevertheless, the compliance matrix of the planar spring keeps invariant.

$$\mathbf{K}_0' = \mathrm{Ad}_{g0}\mathbf{K}_0\mathrm{Ad}_{g0}^T \tag{4-17}$$

where $\mathrm{Ad}_{g0} = \begin{bmatrix} \mathbf{R}_y(\varphi) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_y(\varphi) \end{bmatrix}$, and it is a 6×6 matrix representation of the special Euclidean group *SE*(3), which means that $\mathrm{Ad}_{g0}^T = \mathrm{Ad}_{g0}^{-1}$. Hence, the congruence transformation in (4-17) is equivalent to a similarity transformation of $\mathbf{K}_0$. Considering that $\mathbf{K}_0$ is a diagonal matrix, therefore it is an invariant with respect to the similarity transformation, i.e. $\mathbf{K}_0' = \mathbf{K}_0$. This reveals the isotropism of this planar spring.

The above proves that this type of planar spring displays the isotropic rotational compliance, which means that a moment about any lines passing through the origin in the *xz*-plane of the global coordinate frame {*xyz*} produces the same angular displacement. This is an attractive characteristic in considering tendon channel arrangements and multiple groups of tendon path for multi-segment continuum manipulator. Equation (4-17) is valid for any planar spring configurations that have the diagonal stiffness matrix, and we verify that four-leg and six-leg symmetric configurations possess such isotropic rotational compliance as well.

In the following, we use a numerical example to further reveal the information embodied in the compliance matrix. The dimensions of the planar spring of our prototype are l = 8mm, b = 1.2mm, h = 1mm, d = 2mm, r = 3.5mm. Polyethylene (Young's module E = 1.1GPa and Poisson's ratio v = 0.42) is

selected as fabrication material (as tabulated in Table 4-1) for use in the example, thus deriving each element of the corresponding numerical compliance matrix.

Table 4-1 Numerical example of compliance elements.

Compliance element	***Polyethylene***
c_{11}	6.23×10^{-5} m/N
c_{22}	3.28×10^{-4} m/N
c_{33}	6.23×10^{-5} m/N
c_{44}	9.54 rad/(N m)
c_{55}	2.71 rad/(N m)
c_{66}	9.54 rad/(N m)

By analysing the numerical results, we can draw the following conclusions.

1. In the group of translational compliance elements (c_{11}, c_{22} and c_{33}), the vertical compliance element c_{22} is about 5 times larger than both the horizontal compliance element c_{11} along x-axis and the horizontal compliance element c_{33} along z-axis. This result agrees with our intuition and the qualitative study by Howell et al. (Parise, Howell, & Magleby, 2001). Such translational motion along the vertical axis of the planar module has been investigated for use in many applications, such as a pneumatic valve controller for Flowserve (Parise, Howell, & Magleby, 2001) and a force sensor (Ataollahi, Fallah, Seneviratne, Dasgupta, & Althoefer, 2014).
2. In the group of rotational compliance elements (c_{44}, c_{55} and c_{66}), the rotational compliance elements both c_{44} and c_{66} about the horizontal x-and z-axes are more than 3 times larger than the rotational compliance element c_{55} about the vertical y-axis, indicating its potential to be used for bending motions in continuum manipulator, while resisting in-plane rotations.

Overall, c_{22}, c_{44} and c_{66} are the major compliance elements. Thus, reasonably, further analysis can focus on the major displacements δ_y, θ_x and θ_z that are produced by the loads f_y, m_x and m_z, respectively.

4.1.4 Varying angles of leg arrangement

This section investigates the variable flexibility of the constructed continuum manipulator. The symbolic formulation of a six-dimensional compliance matrix for planar springs has been given in the last section, and in this section one extra variable – angle of leg arrangement – is made to be involved in the compliance matrix, then the corresponding compliance characteristics of different planar spring configurations are analysed.

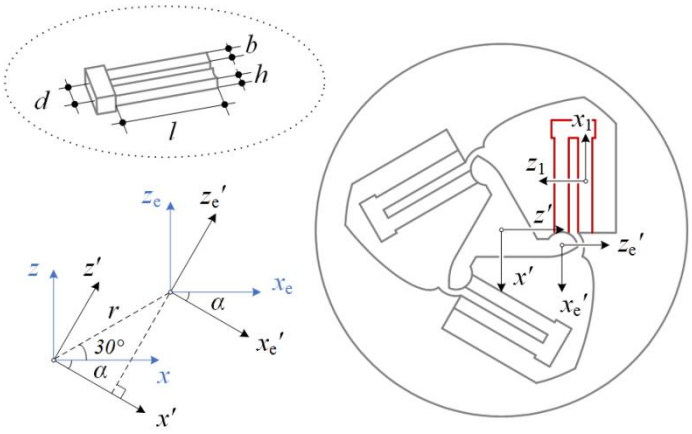


Figure 4-2 Coordinate frames on the planar spring are shown in a 2D front view with the *x*- and *z*-axes illustrated and the *y*-axes complete the right handed frames. All the *x*-axes are in parallel with each other. A zoom-in version of one U-shape leg is given in an isometric view and the corresponding design parameters of the planar spring are labelled. The single leg's global coordinate frame $\{x_e'y_e'z_e'\}$ and the planar spring's global coordinate frame $\{x'y'z'\}$ are described separately in details. The notations without the prime, " ′ ", present the coordinate frames for a side-leg design configuration for comparison.

Equation (4-3) is on the basis of summing up the former researchers' work (von Mises, 1924), (Dai, 2014) and it provides the foundation for deriving the compliance matrix of this type of planar springs. With a bottom-up approach, the compliance matrix of the U-shape leg is then derived and the compliance matrix of the overall planar spring follows. The key for this part of derivation is the coordinate transformation of (4-3).

As the utilised convention in the last section, another coordinate frame $\{x_e'y_e'z_e'\}$ is established at the connecting edge between the leg and the platform, where force is applied to the U-shape leg from the platform. Two flexible beams of the leg have the same compliance matrix expressed in their corresponding coordinate frames, i.e. the centroid of each beam. To shift each of them into coordinate frame $\{x_e'y_e'z_e'\}$, the

adjoint action of Lie group *SE*(3) on its Lie algebra is utilised. The detailed derivation can be found in 4.1. This compliance matrix of the U-shape leg expressed in frame $\{x_e'y_e'z_e'\}$ is written as (4-14).

The global coordinate frame $\{x'y'z'\}$ of the planar spring is established in the centre of the central platform (see Figure 4-2). The distance between the origin of the coordinate frame $\{x'y'z'\}$ and the origin of the coordinate frame $\{x_e'y_e'z_e'\}$ is fixed and denoted by r. The varying angle (α) of leg arrangement results in different configurations of the titled-leg planar springs, and it in the meantime affects the compliance characteristics of the planar springs. The compliance matrix of the whole planar spring is derived and written as

$$\mathbf{C}' = diag\begin{bmatrix} c_{11}' & c_{22}' & c_{33}' & c_{44}' & c_{55}' & c_{66}' \end{bmatrix} \tag{4-18}$$

$$c_{11}' = c_{33}' = \frac{l^3\left(Ad^2 + 4I_y\right)}{3EI_y\left(3Ad^2 + Al^2 + 12I_y\right)}$$

$$c_{22}' = \frac{l\left(3EI_z d^3 + GJl^2\right)}{18EI_z GJ}$$

$$c_{44}' = c_{66}' = \frac{4}{3}\cdot\frac{l\left(EI_z 3d^3 + GJl^2\right)}{EI_z GJ\left(12r^2 + 6d^2 + 4l^2 + 12dr\sin(\frac{\pi}{6}+\alpha) - 12lr\cos(\frac{\pi}{6}+\alpha)\right) + \left(EI_z\right)^2 3d^2 + \left(GJ\right)^2 l^2};$$

$$c_{55}' = \frac{2}{3}\cdot\frac{l^3\left(Ad^2 + 4I_y\right)}{EI_y\left(A\left(5d^2l^2 + (12d^2r^2 + 4l^2r^2 - 12d^2lr)\cos(\frac{\pi}{6}+\alpha)^2 - 6\sqrt{3}d^2lr + 4\sin(\frac{\pi}{6}+\alpha)dl^2r\right) + I_y\left(16l^2 + 48r^2\cos(\frac{\pi}{6}+\alpha)^2 - 48\cos(\frac{\pi}{6}+\alpha)lr\right)\right)}$$

The above compliance matrix reveals to be a diagonal matrix again, like that of the rectangular beam at the beginning. This further simplifies the compliance analysis with the injective function between the applied wrench and the twist deflection. The angular displacement and the translational motion of the platform part are well decoupled. Each element of the diagonal matrix in (4-18) is determined by both the material parameters of the U-shape leg and the mechanical design parameters. Particularly, it is noted that the first three translational compliance elements are invariant to the angle of leg arrangement, while the expressions of the rotational compliance elements contain the variable α that effects the corresponding compliance characteristics.

3.2 Numerical analysis of the compliance characteristics

So far, the six-dimensional compliance matrix of the planar springs with varying angle of leg arrangement is derived. Both of the side-leg design and radial-leg design can also be regarded as the special cases of the titled-leg design and utilised (4-18) to obtain the compliance matrix with the angle $\alpha = 0^\circ$ and $\alpha = 150^\circ$, respectively.

In the following, three numerical examples, see Table 4-2, are given to compare the compliance characteristics of three configurations of planar springs with different angles of leg arrangement.

Table 4-2 Numerical examples for comparison.

Compliance element	***α = 0˚ (side-leg design)***	***α = 90˚***	***α = 150˚ (radial-leg design)***
$c_{11} = c_{33}$	8.31×10^{-5} m/N	8.31×10^{-5} m/N	8.31×10^{-5} m/N
c_{22}	4.37×10^{-4} m/N	4.37×10^{-4} m/N	4.37×10^{-4} m/N
$c_{44} = c_{66}$	61.41 rad/(N m)	17.77 rad/(N m)	15.61 rad/(N m)
c_{55}	6.10 rad/(N m)	2.13 rad/(N m)	4.38 rad/(N m)

In these three examples, the leg design parameters are the same (leg length l = 6mm, width b = 0.9mm, thickness h = 0.75mm, distance d = 1.5mm, radius r = 4mm, please see Figure 4-2.); the fabrication material is chosen as polyethylene (E = 1.1GPa and v = 0.42). Based on the numerical comparisons, it is revealed the compliance elements c_{22}, c_{44}, c_{66} being the main compliance elements whose values are much larger than other elements in the translational compliance element group and in the rotational compliance element group, respectively. The value of c_{22} is more than 5 times larger than that of the element c_{11}or c_{33}; the ratio is varying in the group of the rotational compliance elements ranging from 10 ($\alpha = 0^\circ$) to 4 ($\alpha = 150^\circ$). Therefore, the following analysis will only focus on these three main elements. The values of the translational compliance elements in all the three configurations are the same, while the value of the elements c_{44} and c_{66} decreases with the change of the angle of leg arrangement from 0˚ (side-leg design) to 150˚ (radial-leg design). The results indicate that the side-leg design is more suitable to be used for a flexible arm which is mainly expected to bend during its usage. Furthermore, the diameter of the planar spring is positively correlated with the angle of the leg arrangement, which leads to the design of the conic variable-flexibility continuum manipulator.

Figure 4-3 illustrates the relationship between angle of leg arrangement and rotational compliance in compliance elements c_{22}, c_{44}, c_{55}. The change of the values of compliance elements c_{22} and c_{44} with respect

to rotation angle from angle 0˚ to 90˚ is significant, and therefore the conic variable-flexibility manipulator design will integrate different configurations with the angle varying in this range. Besides, the value of the main compliance elements c_{22} and c_{44} are also reasonable much larger (×8.4) than that of the in-plane rotational compliance element c_{55} which is to be prevented in practical for stability concern.

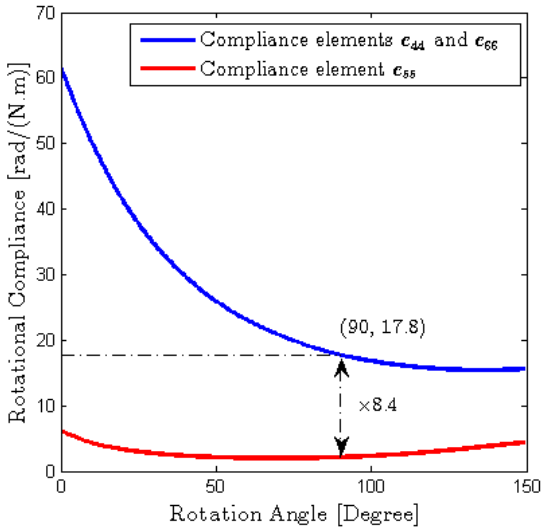


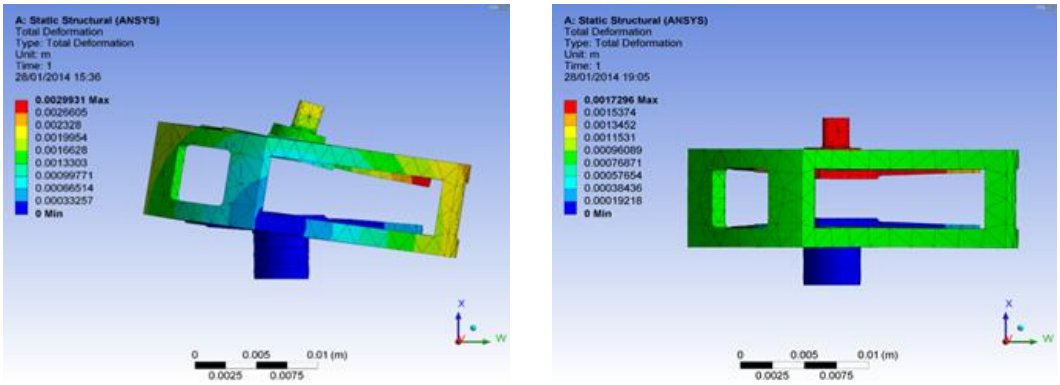
Figure 4-3 Curves illustrating the rotational compliance changes with respect to the rotation angle of leg arrangement. The leg design parameters are the same with the values utilised in the above numerical analysis.

4.1.5 Finite element analysis

FEA simulation was conducted using the commercial software package (ANSYS® 12.0.1 Release). All the important features of deformation of the part are obtained in a static structural analysis environment and the large deflection solver is selected, to ensure that we approach the real behaviour as closely as possible. The numerical simulation results are then compared with the analytical model derived in the beginning of this chapter. The geometry of a double-layer modular segment model was imported with a SolidWorks® geometrical part. The local ANSYS® mesh generator is used to discretize the geometry into 7800 elements and 15459 nodes. The same dimensions and material data were assigned to this simulation study as we noted for the calculations in the previous numerical example in Table 4-1.

Two types of loads were applied to one layer's central platform whilst fixing another layer's central platform. The rotational deflection generated by a moment about horizontal axis is illustrated in Figure 4-4(a); the longitudinal displacement of the segment generated by a force along the vertical axis is

illustrated in Figure 4-4(b). We can see that two compliant layers come into both types of deformations in mirror symmetry.



(a) (b)

Figure 4-4 FEA simulations of the double-layer module. (a) the rotational deflection generated by a moment about horizontal axis; (b) the longitudinal displacement of the segment generated by a force along the vertical axis.

Figure 4-5 and Figure 4-6 illustrate the simulation results for two types of these applied loads to the upper layer's central platform. The line chart of Figure 4-5 depicts the moment-rotation curves of both simulated one layer and double-layer module rotational deflections, where it validates the double-layer module rotation angle doubles the single layer rotation angle at the same amount of moment. Besides, the line curves indicate that the linear relationships are applicable for the designed module in the range of simulated rotation. Combining the analytical curve of a single layer model that was previously derived, we can also see that the results agree well with those correspondingly obtained in FEA simulation. The line chart of Figure 4-6 indicates the same conclusion in the case of longitudinal displacements. Regarding the minor discrepancy between the analytical prediction and the simulation result, we suspect that it originates from the connection part (intermediate platform, see Figure 3-3(a)) between the two parallel beams of each U-shape leg. Because the intermediate platform ideally should be fully rigid, while this part built in FEA simulation possesses the same material properties as rest of the module.

In summary, the double-layer compliant module demonstrates an ideal simulation of rotational deflections and longitudinal displacements, which simultaneously can be well predicted with a screw-theory based analytical model. This simulation verification assures that there are no additional unmodeled

effects in the components of the structure that are contributing significantly to the overall system compliance.

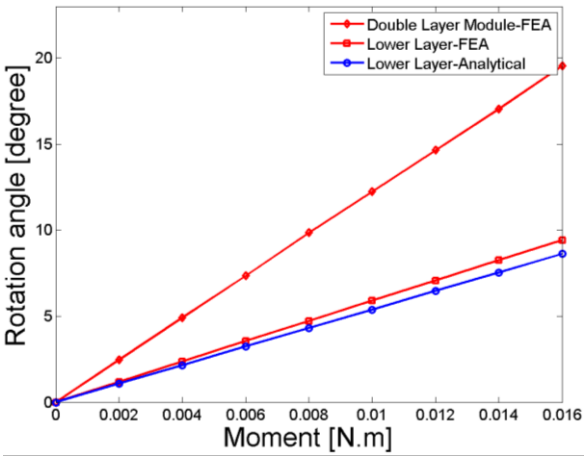


Figure 4-5 Moment-rotation diagram for the double-layer compliant module via analytical model predication and FEA simulation results.

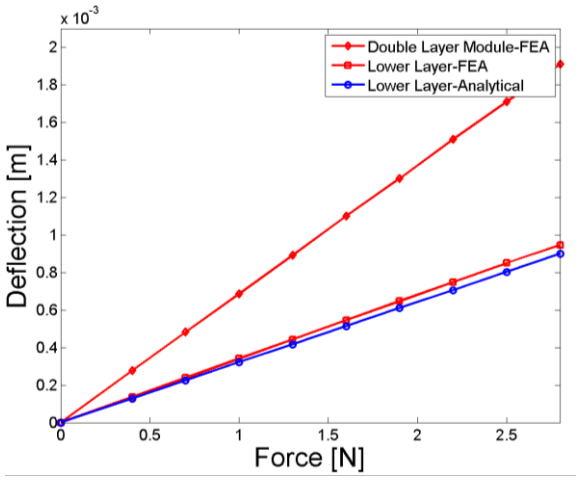


Figure 4-6 Force-deflection diagram for the double-layer compliant module via analytical model predication and FEA simulation results.

So far, the first part of the Finite Element Analysis (FEA) simulations conducted on double-layer modules using ANSYS® have validated the developed six-dimensional compliance matrix of (4-16). In the following, FEA simulations were conducted on various configurations of the planar spring to reveal the differences of their compliance properties. The numerical simulation results are then compared with the analytical model derived as in (4-18). Similar with previous simulations, polyethylene is assigned to the simulated material in these cases. And also two types of loads are applied to the platform whilst fixing the

base of the planar spring. In Figure 4-7, the rotational deflection (Figure 4-7(a)) and the longitudinal displacement (Figure 4-7(b)) of one example of the planar spring configuration are illustrated.

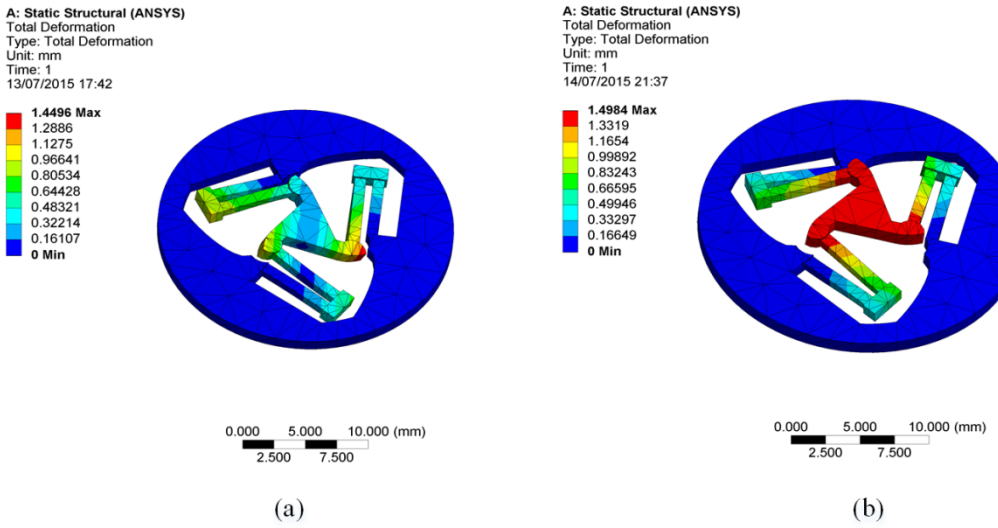


Figure 4-7 FEA simulations of a titled-leg planar spring ($\alpha = 30^{\circ}$). Two types of loads (a): moment; (b): force, are applied to the platform whilst fixing the base of the planar spring.

Plots representing the numerical results based on the FEA simulations are shown in Figure 4-8 and Figure 4-9. Four different configurations of the titled-leg planar springs are studied and the corresponding angle of leg arrangement takes the value $\alpha = 0^{\circ}$, 30°, 60°, and 90°, respectively. The FEA results are compared to analytical prediction results based on the developed compliance matrix of (4-18). In moment-rotation plots of Figure 4-8, four groups of FEA simulation and analytical prediction curves corresponding to different configurations of planar springs are separated from each other, demonstrating different rotational compliance characteristics. Figure 4-9, four FEA simulation curves are almost fitted together, while their analytical model, the compliance element c_{22} in (4-18) predicts the same result. These comparison results are consistent with actual situation. Strong correlation FEA simulation and analytical model is revealed, although there are slight discrepancies. We suspect that the reason for the discrepancy comes from the following aspect. It is that the whole planar spring part when "built" in the FEA modeller possesses the same material properties, thus there is no distinction between flexible leg and other rigid connection parts. In conclusion, the FEA simulation further indicates the varying rotational compliance of planar springs with respect to different angles of leg arrangement and validates the analytical model of the main compliance elements in (4-18).

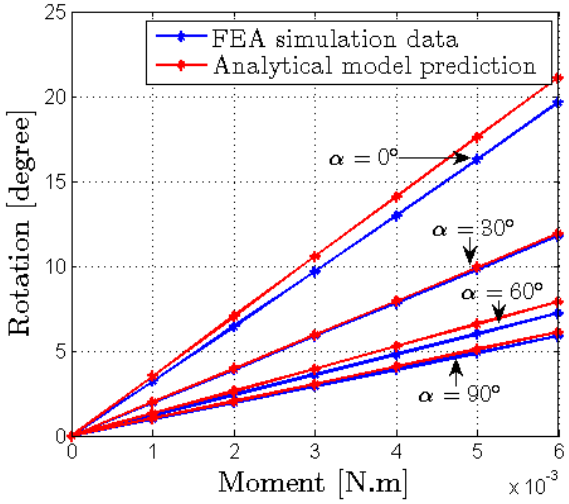


Figure 4-8 Numerical comparison of the analytical model and the FEA simulation. Four different configurations (α = 0˚, 30˚, 60˚, and 90˚) of planar springs are investigated. The diagrams represent the out-plane rotational deflection results

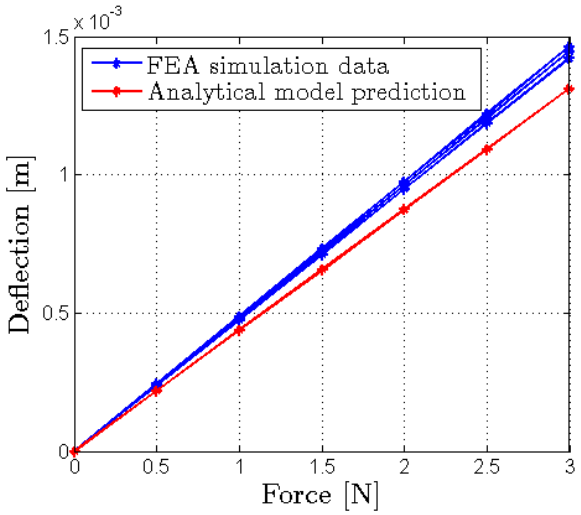


Figure 4-9 Numerical comparison of the analytical model and the FEA simulation. Four different configurations (α = 0˚, 30˚, 60˚, and 90˚) of planar springs are investigated. The diagrams represent the out-plane translational displacement of the central platform.

4.2 Analysis on the continuum manipulator

A helical spring is commonly used to provide the axial displacement proportional to an applied force in the same direction according to Hooke's law. In this thesis, under the framework of the generalised Hooke's spring law, we analyse another type of spring – planar spring, and for the first time utilise their both translational and angular motions to generate the bending and/or contraction deflections of the continuum manipulator. Before proceeding to the following analysis, there arise two assumptions. One is that the loads exerted on the top plate of the robot are uniformly distributed to each compliant layer. The other is that the frictional effects of tendons are neglected. In the literature, some local spring-based models for continuum robots have been proposed previously in (Giri & Walker, 2011), (Jung, Penning, Ferrier, & Zinn, 2011).

4.2.1 Compliance of the robot

The serially-connected assembly of multiple layers of planar springs has the compliance attribute equal to the sum of compliance matrix of each layer of the planar spring, i.e. theoretically, $\mathbf{C} = n\mathbf{C}_0$, where $\mathbf{C}$ represents the equivalent compliance of the continuum manipulator in form of a 6×6 diagonal matrix, and n is the total number of planar spring layers. On the other hand, as we discussed in the end of 4.1.3, among the total six compliance elements in (4-16), only major compliance elements c_{22}, c_{44} and c_{66} are of our interest. The rest of compliance elements are at relatively small values and can be neglected (Su, Shi, & Yu, 2012). Thus, the effectual compliance matrix of the continuum manipulator is

$$\bar{\mathbf{C}} = n\bar{\mathbf{C}}_0 = diag\left[n \cdot c_{22} \quad n \cdot c_{44} \quad n \cdot c_{66}\right] \tag{4-19}$$

where $\bar{\mathbf{C}}, \bar{\mathbf{C}}_0$ respectively denotes the compliance matrix that only contains three major compliance elements of the robot and a planar spring.

With (4-19), the following relation between end-point force and robot deflection holds.

$$\begin{bmatrix} \Psi_y \\ \Omega_x \\ \Omega_z \end{bmatrix} = \bar{\mathbf{C}} \begin{bmatrix} F_y \\ M_x \\ M_z \end{bmatrix} = \begin{bmatrix} n \cdot c_{22} & 0 & 0 \\ 0 & n \cdot c_{44} & 0 \\ 0 & 0 & n \cdot c_{66} \end{bmatrix} \begin{bmatrix} F_y \\ M_x \\ M_z \end{bmatrix} \tag{4-20}$$

where $[F_y, M_x, M_z]^T$ represents the effective Cartesian force-moment vector on the end-point of the robot; Ω_x, Ω_z are the two rotation angles of the moving frame $\{xyz\}$ attached to the end-point of the continuum manipulator, illustrated in Figure 4-10(a); particularly, Ψ_y is the change of the total length of the continuum manipulator.

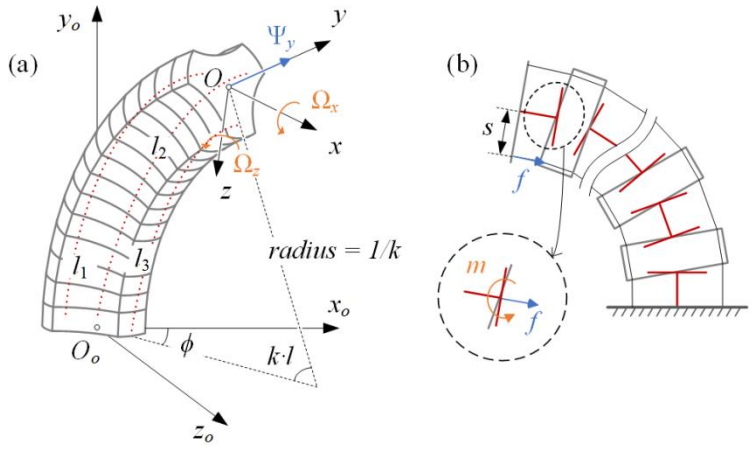


Figure 4-10 (a) Configuration of a bending continuum manipulator and its attached coordinate frames; (b) 2D view of the bending configuration; the zoomed-in view shows the equivalent force and moment acting on the last module when a pulling force is exerted to the top plate via a tendon.

4.2.2 Discussions on bending deflection decoupled with contraction

There are two factors giving rise to the decoupling property that the robot effectively decreases the contraction when generating bending deflections. One is that the compliance matrix in (4-16) is diagonal. Otherwise, if the entries outside the main diagonal are not all zero, the applied moment intended for the bending deflection will simultaneously generate the contraction (or the compression force will result in bending).

The other reason is that the rotational compliance has much more significant effects on the robot deflection than the translational compliance. This can be clarified as follows. Assuming a pulling force f exerted to the distal plate via the tendon (please refer to Figure 4-10(b)), this load is equivalent to a moment $m = f \cdot s$, where s is the lever-arm distance to produce the moment of force, and a force f acting on upper layer platform of the last module and further being transmitted to the full robot. According to the previous compliance analysis, this moment on the robot will generate a total angular displacement $\Omega = n \cdot c_r \cdot f \cdot s$, and this force will generate a translational displacement $\Psi = n \cdot c_t \cdot f$. ($c_r$ and c_t denote the rotational compliance and translational compliance; referring to Table 4-1, $c_r = 9.54$rad/(N·m), $c_t = 3.28 \times 10^{-4}$m/N; in the current robot design, $s = 13.4$mm, and the total number of planar springs $n = 20$.) Thus, if the exerted force $f = 1$N, we can calculate $\Omega = 146.5$degree and $\Psi = 0.328$mm (the total length of the robot is 143mm).

This numerical comparison reveals that when the robot bends to a very large angle, the total contraction of the robot almost remains the same.

Both of the above discussed factors origin from the properties of the planar spring. Other kinds of compliant modules, such as helical springs (Yang, Jason, & Abdel-Malek, 2006) and pneumatic actuators (McMahan, et al., 2006), (Kier & Smith, 1985) as reviewed in Chapter 3, do not possess this feature that makes the bending deflection of this continuum manipulator effectively decoupled from the contraction.

4.2.3 Simultaneous rotations

The two rotations Ω_x, Ω_z are relative to the moving frame $\{xyz\}$ attached to the end-point of the continuum manipulator. We can observe that the rotation axes of Ω_x, Ω_z are not fixed, which makes it complicated to analyse the movements of the continuum manipulator. In order to address this matter, we define a unit vector $\boldsymbol{u} = \left[u_x, u_y, u_z \right]^T$ in the reference coordinate frame $\{x_o y_o z_o\}$, to represent spatial direction from the origin O_o to the origin O of the moving frame (please refer to Figure 4-11(a)). Thus, instead of describing the bending of the continuum manipulator with two rotations Ω_x, Ω_z, we can examine the rotation of the vector $\boldsymbol{u}$, which can be described by another two rotations Φ_x, Φ_z about the x-axis and the z-axis of the fixed reference frame $\{x_o y_o z_o\}$. In addition, a half-angle relation holds between the rotations of the distal moving frame and the corresponding rotations of the vector u (please refer to Figure 4-11(b)), i.e. $\Phi_x = \Omega_x/2$, $\Phi_z = \Omega_z/2$.

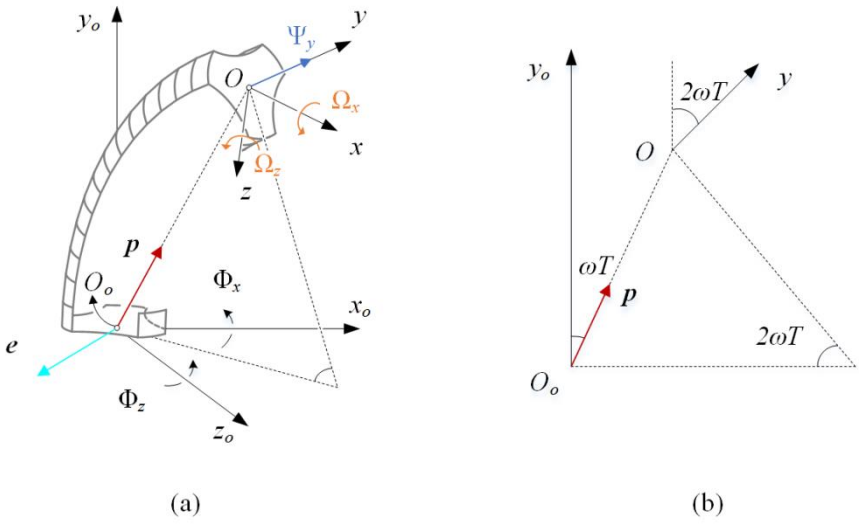


Figure 4-11 (a) Illustration of simultaneous rotations and the unit vector $\boldsymbol{u}$ and rotation axis $\boldsymbol{e}$. (b) the bending plane geometry.

Considering the synchronicity of the rotation Φ_x and the rotation Φ_z, we introduce angular velocities ω_x and ω_z respectively and we assume that both of them are constant finite velocities (in this case, angular velocity ω_y is equal to zero for the actual physical conditions). We further represent angular velocities as vectors pointing along the axis of rotation

$$\boldsymbol{\varpi}_x = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}; \boldsymbol{\varpi}_y = \begin{bmatrix} 0 \\ \omega_y \\ 0 \end{bmatrix}; \boldsymbol{\varpi}_z = \begin{bmatrix} 0 \\ 0 \\ \omega_z \end{bmatrix} \tag{4-21}$$

Then, we can add angular velocities by vector addition as (Beatty, 1986)

$$\boldsymbol{\varpi} = \boldsymbol{\varpi}_x + \boldsymbol{\varpi}_y + \boldsymbol{\varpi}_z = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{4-22}$$

It is important to note that the addition operation only holds for angular velocity vectors, while it cannot for rotations (Beatty, 1986).

The angular velocity vector of (4-22) specifies a velocity quantity with a unit vector $\boldsymbol{e}$ (passing through the origin O_o of the reference coordinate system) indicating the direction of an axis of rotation and the scalar ω indicating the angular velocity. Thus, we can also write

$$\boldsymbol{\varpi} = \omega \boldsymbol{e} \tag{4-23}$$

where it conforms to the right-hand rule.

As we have assumed that the angular velocity is constant (due to the fact that in our situation the bending deformation is low speed motion), multiplying an observation interval T to the angular velocity ω, we can derive that the rotation action of the unit vector $\boldsymbol{u}$ about the rotation axis $\boldsymbol{e}$ is equal to ωT – hence, we realize the concept of simultaneous rotation. An alternative derivation is also achieved via infinitesimally small rotations (Stancin & Tomazic, 2011)

$$\boldsymbol{e} = \frac{1}{\sqrt{\omega_x^2+\omega_y^2+\omega_z^2}}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} = \frac{1}{T\sqrt{\omega_x^2+\omega_y^2+\omega_z^2}}\begin{bmatrix}\omega_x T\\ \omega_y T\\ \omega_z T\end{bmatrix} = \frac{1}{\sqrt{\Phi_x^2+\Phi_y^2+\Phi_z^2}}\begin{bmatrix}\Phi_x\\ \Phi_y\\ \Phi_z\end{bmatrix} \tag{4-24}$$

$$\omega T = T\sqrt{\omega_x^2+\omega_y^2+\omega_z^2} = \sqrt{\Phi_x^2+\Phi_y^2+\Phi_z^2} \tag{4-25}$$

Given an axis $\boldsymbol{e}$ and a rotation angle ωT, we can rotate the unite vector $\boldsymbol{u}$ via Rodrigues' rotation formula (Tsai, 1999), (Dai, 2006), (Dai, 2014)

$$\mathbf{R} = \cos\beta\mathbf{I} + \sin\beta\hat{\boldsymbol{s}} + (1-\cos\beta)\boldsymbol{s}\boldsymbol{s}^T \tag{4-26}$$

where $\mathbf{I}$ is a 3×3 identity matrix; β specifies the rotation angle which is equal to ωT; $\boldsymbol{s}$ denotes the axis about which the vector is rotating. The Rodrigues' rotation formula constructs a rotation matrix $\mathbf{R}$ to rotate a vector with the rotation β ($-\pi<\beta<\pi$) about the axis $\boldsymbol{s}$.

Multiplying the rotation matrix $\mathbf{R}$ on the left of unit vector $\left[0,0,1\right]^T$ which indicates the initial position of the direction vector $\boldsymbol{u}$, the resulting orientation is given by

$$\boldsymbol{u} = \mathbf{R}\begin{bmatrix}0\\ 0\\ 1\end{bmatrix} \tag{4-27}$$

Then, we can calculate the distance from the origin O_o to the origin O based on the geometrical relationship and derive the position vector of the origin O in the reference frame.

4.2.4 Overview of continuum manipulator kinematics and statics

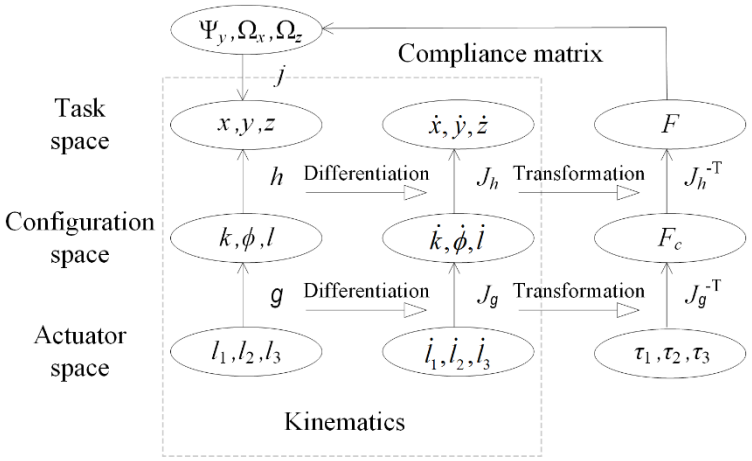


Figure 4-12 An overview of kinematic and static relations of the continuum manipulator. The variables in each ellipse represent the chosen mathematical description with respect to the proposed three-tendon driven robot in this paper. l_1, l_2, l_3 are three tendon lengths; k, ϕ, l are the configuration space variables (k is curvature; ϕ is bending angle; l is the total length; all the three variables are frequently used in references, all are defined in Figure 4-10(a)). x, y, z denote the tip position of the continuum manipulator in Cartesian space. F_c is a three-dimensional vector and its elements are related to the corresponding configuration variables. F is the effective Cartesian force-moment vector and is represented as the vector $[F_y, M_x, M_z]^T$ in (4-20). The notation $(\cdot)$ denotes the time-derivative.

We put our design of this continuum manipulator in the context of continuum robotics. Kinematics and statics of such robots have been intensively studied (Xu & Simaan, 2010), and review papers can be referred to (Walker, 2013), (Webster & Jones, 2010). As for our three-tendon driven continuum manipulator in this paper, an overview is summarized in Figure 4-12, to clarify kinematic relation between tendon lengths and task space variables, and static force relation between tendon tensions and end-point force. For the detailed functions of the kinematic mappings listed in Figure 4-12, i.e. sub-mappings h, g, and their corresponding Jacobians $\mathbf{J}_h$, $\mathbf{J}_g$, they will be presented in the following section, and please also refer to (Walker, 2013), (Webster & Jones, 2010), (Hannan & Walker, 2003), where the same notational convention is employed. Besides, the mapping j is derived with the concept of simultaneous rotation and the use of Rodrigues' formula and it is discussed in 4.2.3. The derivation of the static force analysis associated with $\mathbf{J}_g^{-T}$ and $\mathbf{J}_h^{-T}$ is based on the principle of virtual power.

The kinematic modelling complies with the constant-curvature approximation (Webster & Jones, 2010) and thus the configuration of the backbone can be described by 3D arc parameters, as given in Figure 4-12.

In this regards, the static analysis will treat the robot as an elastic unit only with three independent DOFs – two rotational DOFs, Ω_x, Ω_z, and one translational DOF, Ψ_y. Here, the robot does not behave as an arbitrarily deformable structure with infinite degrees of freedom (Walker, 2013), (Webster & Jones, 2010).

Although the robot is supposed to be in three-tendon actuation manner, one tendon actuation still enables a simple bending deflection, as illustrated in Figure 4-10(b). Three-tendon actuation then increases the rotational DOF from 2D to 3D, and enables the third translational DOF underlying the physical structure. The varying length of the robot results in an enlarged work space, as discussed in the following section. The motion control of the continuum manipulator is commonly conducted by controlling the lengths of tendons to accomplish the tip position tracking tasks.

4.3 Constant-curvature continuum manipulator kinematics

In this section, we provide a brief overview of the general continuum manipulator kinematics using constant-curvature theory. The derived models form the basis for the robot controller development.

4.3.1 Introduction of continuum manipulator kinematics

The constant-curvature arc approximation has been frequently applied to the kinematic modelling of many continuum manipulators (Hannan & Walker, 2003), (Webster & Jones, 2010). Different modelling approaches producing equivalent results of constant-curvature forward kinematics are reviewed and unified in (Webster & Jones, 2010). Due to its simplifications in modelling, it enables an analytical closed-form relationship between actuator inputs and arc parameters useful for real-time control. Extensions of fundamental concept of constant-curvature kinematics, piecewise constant-curvature and finite-fragmentation curvature modelling are proposed to fit the physical model of manipulators with a multi-section backbone or a variable section curvature (Mahl, Hildebrandt, & Sawodny, 2014). The latter considers the backbone shape comprised of a finite number of small curved units and it is equivalent to modelling a single section with piecewise constant-curvature approximation. Unlike conventional rigid-link discrete manipulators (Craig, 2005) the use of joint variables and link parameters does not directly yield continuum kinematics. The elastic bending feature of a continuum manipulator leads kinematics to be decomposed into two submappings that link together with configuration variables (please refer to Figure 4-13). In order to lighten the notation, throughout the paper, we drop all time-varying variable notations.

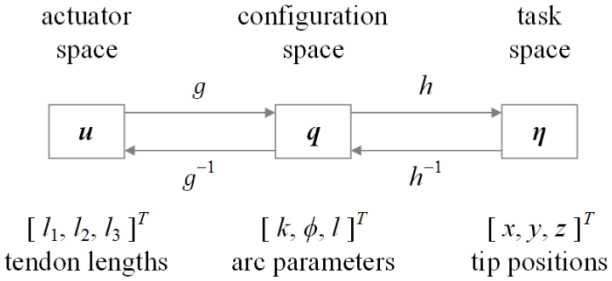


Figure 4-13 Kinematic mapping and its decomposition of a continuum manipulator modelled using constant-curvature theory. The manipulator-specific submapping g relates actuator space variables $\boldsymbol{u}$ and configuration space variables $\boldsymbol{q}$; the manipulator-independent mapping h relates configuration space variables $\boldsymbol{q}$ and task space variables $\boldsymbol{\eta}$. g^{-1} and h^{-1} denote the inverse mappings of each submappings, respectively. In this case, tendon-driven actuation manner is applied and three tendon lengths $[\, l_1, l_2, l_3 \,]^T$ represent the actuator variables; arc parameter under the constant-curvature approximation is $[\, k, \phi, l \,]^T$; only tip position $[\, x, y, z \,]^T$ is taken to represent the pose states.

The two decomposed submapping portions are respectively described by a manipulator-specific kinematics $g : \mathbb{R}^n \to \mathbb{R}^n, \boldsymbol{u} \mapsto \boldsymbol{q}$, and a manipulator-independent kinematics $h : \mathbb{R}^n \to \mathbb{R}^n, \boldsymbol{q} \mapsto \boldsymbol{\eta}$. The former varies with different actuation manners (although sometimes there exists certain correlation among common actuation strategies), while the latter is totally general and applies to all the individual sections of a continuum manipulator under the assumption of constant-curvature. Hence, the complete kinematics mapping that computes the end-effector's pose $\boldsymbol{\eta}$ depending on the actuator state $\boldsymbol{u}$ is given by

$$C_h g = g \circ h \tag{4-28}$$

where C_h is a composition operator defining a chaining process where the output of the function g becomes the input of the function h.

Without limiting generality, the actuator space variables are chosen as the most direct actuation – tendon-driven design, where an arc is shaped by tendons. Herein, the three tendon lengths are written in vector form as follows

$$\boldsymbol{u} = \left[l_1, l_2, l_3 \right]^T \tag{4-29}$$

The arc parameters are represented by the curvature (k), rotational angle (ϕ), and arc length (l). (please refer to Figure 4-14). The configuration space triplets above are all functions of the actuator variables, i.e.

$$\boldsymbol{q} = \left[k(\boldsymbol{u}), \phi(\boldsymbol{u}), l(\boldsymbol{u}) \right]^T \tag{4-30}$$

Furthermore, the arc geometry provides the relationships $\theta = k\,l$ and $r = 1/k$ which enables the calculation of the arc bending angle θ and radius r. Regarding the pose representation, we only specify the position of the end-effector for the purpose of motion control, which in three-dimensional Euclidean space is defined by the vector

$$\eta = \left[x, y, z\right]^T \tag{4-31}$$

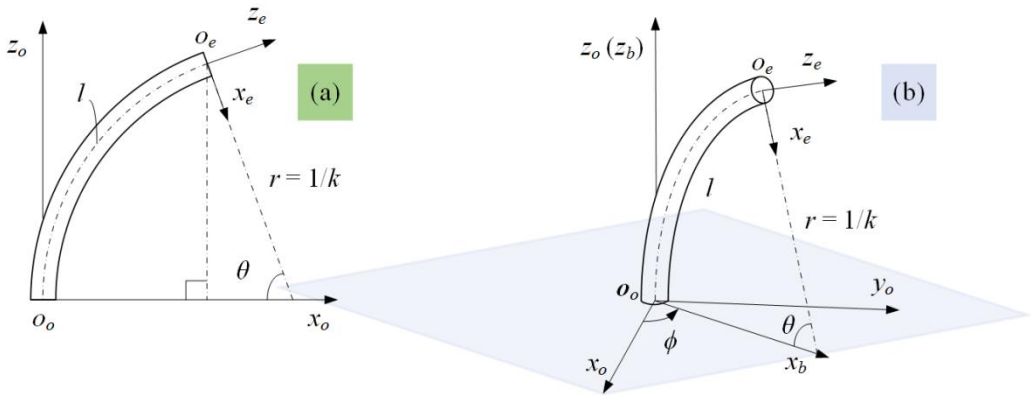


Figure 4-14 Diagram of a continuum manipulator bending in (a) 2D space and in (b) 3D space. The configuration variables and different coordinate systems are illustrated.

4.3.2 Coordinate systems

With different coordinate frame choices, the derived kinematic mapping would be diverse in form. In order to describe the position of the end-effector in the universe, the reference coordinate system must be first established. Likewise, additional moving frames attached to the continuum manipulator are introduced. All the coordinate systems with respect to a single-section continuum manipulator (please refer to Figure 4-14) are described below.

1. *Reference Coordinate System* $\{x_o y_o z_o\}$, for convenience, is fixed to the proximal end of the continuum manipulator with its z_o axis tangent to the backbone curve of the bending manipulator and pointing toward the distal end. The $x_o y_o$ plane is perpendicular to the bending plane.
2. *Bending Plane Coordinate System* $\{x_b y_b z_b\}$ is defined such that continuum manipulator always bends in the $x_b z_b$ plane. The origin o_b coincides with the origin o_o and the axis z_b is collinear with the axis z_o.
3. *End-effector Coordinate System* $\{x_e y_e z_e\}$ is attached to the tip of the continuum manipulator. The origin o_e is at the centre of the tip cross section and the z_e axis tangent to the backbone curve, or

equivalently normal to the tip cross section. For convenience, the $x_e z_e$ plane is coplanar with the bending plane $x_b z_b$.

4.3.3 Manipulator-independent submapping

Once the aforementioned coordinate systems are established, the problem of deriving the manipulator-independent submapping is transformed into solving the mathematics of mapping to describe the end-effector coordinate system $\{x_e y_e z_e\}$ relative to the reference coordinate system $\{x_o y_o z_o\}$. Thus, a parameterised homogenous transformation can be used as

$$ {}^{o}\mathbf{T}_{e}(\boldsymbol{q}) = \begin{bmatrix} {}^{o}\mathbf{R}_{e}(\boldsymbol{q}) & {}^{o}\boldsymbol{p}_{e}(\boldsymbol{q}) \\ \boldsymbol{0}^{T} & 1 \end{bmatrix} \tag{4-32} $$

where the 4×4 homogeneous transformation matrix ${}^{o}\mathbf{T}_{e}(\boldsymbol{q})$ constitutes the standard representation of the special Euclidean group $SE(3)$ with the effect of transforming the coordinate frame $\{x_e y_e z_e\}$ to the reference coordinate frame $\{x_o y_o z_o\}$; the matrix ${}^{o}\mathbf{T}_{e}(\boldsymbol{q})$ is constructed by a 3×3 rotation matrix ${}^{o}\mathbf{R}_{e}(\boldsymbol{q})$ and a 3×1 position vector ${}^{o}\boldsymbol{p}_{e}(\boldsymbol{q})$; ${}^{o}\mathbf{R}_{e}(\boldsymbol{q})$ is an element of the special orthogonal group $SO(3)$ and denotes the orientation of the coordinate frame $\{x_e y_e z_e\}$ relative to the reference coordinate frame $\{x_o y_o z_o\}$; the 3-vector ${}^{o}\boldsymbol{p}_{e}(\boldsymbol{q})$ is an element of the translation group $T(3)$ and denotes the position of the origin of coordinate frame $\{x_e y_e z_e\}$ relative to the reference coordinate frame $\{x_o y_o z_o\}$.

Each components of the homogeneous transformation matrix are derived as follows. The columns of the rotation matrix ${}^{o}\mathbf{R}_{e}(\boldsymbol{q})$ can be obtained by writing the unit vectors that define directions of the principle axes of end-effector coordinate system $\{x_e y_e z_e\}$ in reference coordinate frame $\{x_o y_o z_o\}$. ${}^{o}\boldsymbol{p}_{e}(\boldsymbol{q})$ is a pure position vector translating the point in space. The operations of $SE(3)$ can be performed through the matrix multiplication, with the transformation composition implemented. Therefore, ${}^{o}\mathbf{T}_{e}(\boldsymbol{q}) = {}^{o}\mathbf{T}_{b}(\boldsymbol{q})\,{}^{b}\mathbf{T}_{e}(\boldsymbol{q})$. Note here that the operations of $SE(3)$ is non-commutative, hence the order for composition is important. Composition of the homogeneous transformation matrix ${}^{o}\mathbf{T}_{e}(\boldsymbol{q})$ is accomplished as

$$ {}^{o}\mathbf{T}_{e}(\boldsymbol{q}) = \begin{bmatrix} {}^{o}\mathbf{R}_{b}(\phi) & \mathbf{0} \\ \boldsymbol{0}^{T} & 1 \end{bmatrix} \begin{bmatrix} {}^{b}\mathbf{R}_{e}(k,l) & {}^{b}\boldsymbol{p}_{e}(k,l) \\ \boldsymbol{0}^{T} & 1 \end{bmatrix} = \begin{bmatrix} {}^{o}\mathbf{R}_{b}(\phi)\,{}^{b}\mathbf{R}_{e}(k,l) & {}^{o}\mathbf{R}_{b}(\phi)\,{}^{b}\boldsymbol{p}_{e}(k,l) \\ \boldsymbol{0}^{T} & 1 \end{bmatrix} \tag{4-33} $$

where one homogeneous transformation matrix describing the frame $\{x_b y_b z_b\}$ relative to the reference frame $\{x_o y_o z_o\}$ is ${}^{o}\mathbf{T}_{b}(\boldsymbol{q}) = \begin{bmatrix} {}^{o}\mathbf{R}_{b}(\phi) & \mathbf{0} \\ \boldsymbol{0}^{T} & 1 \end{bmatrix}$, and another one describing the frame $\{x_e y_e z_e\}$ relative to the frame $\{x_b y_b z_b\}$ is ${}^{b}\mathbf{T}_{e}(\boldsymbol{q}) = \begin{bmatrix} {}^{b}\mathbf{R}_{e}(k,l) & {}^{b}\boldsymbol{p}_{e}(k,l) \\ \boldsymbol{0}^{T} & 1 \end{bmatrix}$.

Referring to Figure 4-14(a), the 2D bending model of a continuum manipulator reveals that the position vector ${}^{b}\boldsymbol{p}_{e}(k, l)$ can be written as

$$ {}^{b}\boldsymbol{p}_{e}(k,l) = \begin{bmatrix} \dfrac{1-\cos(k \cdot l)}{k} \\ 0 \\ \dfrac{\sin(k \cdot l)}{k} \end{bmatrix} \tag{4-34} $$

And the rotation matrix ${}^{b}\mathbf{R}_{e}(k, l)$ represents a rotation of θ (= $k \cdot l$) about the axis y_b and it can be written as

$$ {}^{b}\mathbf{R}_{e}(k,l) = \begin{bmatrix} \cos(k \cdot l) & 0 & \sin(k \cdot l) \\ 0 & 1 & 0 \\ -\sin(k \cdot l) & 0 & \cos(k \cdot l) \end{bmatrix} \tag{4-35} $$

Similarly, the rotation matrix ${}^{o}\mathbf{R}_{b}(\phi)$ representing the bending plane rotation of ϕ about the axis z_o can be written as

$$ {}^{o}\mathbf{R}_{b}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4-36} $$

Therefore, the complete homogeneous transformation matrix ${}^{o}\mathbf{T}_{e}(\boldsymbol{q})$ can be calculated. When in the case only considering the end-effector position representation, the manipulator-independent submapping h only takes the first three elements of the last column of ${}^{o}\mathbf{T}_{e}(\boldsymbol{q})$, i.e. $h = {}^{o}\boldsymbol{p}_{e}(\boldsymbol{q})$.

$$ h = {}^{o}\mathbf{R}_{b}(\phi)\,{}^{b}\boldsymbol{p}_{e}(k,l) = \begin{bmatrix} \dfrac{\cos\phi(1-\cos(k \cdot l))}{k} \\ \dfrac{\sin\phi(1-\cos(k \cdot l))}{k} \\ \dfrac{\sin(k \cdot l)}{k} \end{bmatrix} \tag{4-37} $$

So far, we complete the derivation of the manipulator-independent submapping based on a homogeneous transformation. Furthermore, using the derived 4×4 homogeneous transformation matrix ${}^{o}\mathbf{T}_{e}(\boldsymbol{q})$, any vectors ${}^{e}\boldsymbol{s}$ expressed relative to the end-effector coordinate system $\{x_e y_e z_e\}$ can be expressed relative to the reference coordinate system $\{x_o y_o z_o\}$. Thus,

$$ \begin{bmatrix} {}^{o}\boldsymbol{s} \\ 1 \end{bmatrix} = {}^{o}\mathbf{T}_{e}(\boldsymbol{q}) \begin{bmatrix} {}^{e}\boldsymbol{s} \\ 1 \end{bmatrix} \tag{4-38} $$

Equation (4-38) can be used to solve for the pose representation including end-effector positions and orientations.

4.3.4 Manipulator-specific submapping

Now to find the manipulator-specific submapping, we decide to adopt the three-tendon-driven actuation strategy. First, we assume that all the three tendons are in tension during the manipulator articulation and there is no slack. Referring to the references (Webster & Jones, 2010), (Mahl, Hildebrandt, & Sawodny, 2014), all the defined configuration variables can be expressed with respect to tendon actuation variables as follows

$$k(\boldsymbol{u}) = \frac{2\sqrt{l_1^{\,2} + l_2^{\,2} + l_3^{\,2} - l_1 l_2 - l_1 l_3 - l_2 l_3}}{d\left(l_1 + l_2 + l_3\right)} \tag{4-39}$$

$$\phi(\boldsymbol{u}) = \tan^{-1}\left(\frac{\sqrt{3}(l_2 + l_3 - 2l_1)}{3(l_2 - l_3)}\right) \tag{4-40}$$

$$l(\boldsymbol{u}) = \frac{1}{3}\sum_{i=1}^{3} l_i \tag{4-41}$$

where d is the radius of the cross-section of continuum manipulator and here it is approximately equal to the distance from the centre of the tendon channel to the centre of the cross-section.

This mathematic model is most frequently applied to tendon-driven continuum manipulators and also to any continuously bending actuator, for example, the bellow-like actuators in Festo's Bionic Handling Assistant (BHA) (Festo Corporate, 2010), (Mahl, Hildebrandt, & Sawodny, 2014). More detailed derivations of the above (4-39), (4-40), (4-41) can be found in (Webster & Jones, 2010). Hereby, we complete the manipulator-specific submapping g and end the forward kinematics. Upon the analytical kinematic modelling, the inverse mapping can be further derived and in case of the current simplified model, both the submappings g^{-1} and h^{-1} can be produced analytically by solving the nonlinear equations defined by forward mappings g and h.

4.3.5 Jacobians

The Jacobian is a multidimensional form of partial derivatives with respect to time of the forward kinematics. It reveals the velocity-level forward kinematics that the actuator velocities to the spatial velocity of the end-effector. Given the forward kinematics of the form

$$\boldsymbol{\eta} = f(\boldsymbol{u}) = h\big(g(\boldsymbol{u})\big) \tag{4-42}$$

then, the velocity kinematics is derived based on high dimensional chain rule as

$$\dot{\boldsymbol{\eta}} = \frac{\partial f}{\partial \boldsymbol{u}}\dot{\boldsymbol{u}} = \frac{\partial h}{\partial \boldsymbol{q}}\frac{\partial g}{\partial \boldsymbol{u}}\dot{\boldsymbol{u}} \tag{4-43}$$

This yields the Jacobian matrix equals

$$\begin{gathered}\mathbf{J}(\boldsymbol{u}) = \frac{\partial h}{\partial \boldsymbol{q}}\frac{\partial g}{\partial \boldsymbol{u}} = \mathbf{J}_h(\boldsymbol{q})\mathbf{J}_g(\boldsymbol{u}) \\ \dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{u})\dot{\boldsymbol{u}}\end{gathered} \tag{4-44}$$

where $\mathbf{J}(\boldsymbol{u})$ is a time-varying 3×3 matrix, whose elements are nonlinear functions of instant actuator states expressed by $\boldsymbol{u}$.

In (4-44), the left component of the Jacobian represents the Jacobian $\mathbf{J}_h(\boldsymbol{q})$ of the manipulator-independent portion of kinematics and the right component of the Jacobian represents the Jacobian $\mathbf{J}_g(\boldsymbol{u})$ of the manipulator-independent portion of kinematics. We get both explicit Jacobian matrices as

$$\mathbf{J}_h(\boldsymbol{q}) = \begin{bmatrix} \dfrac{\cos\phi\left(k\cdot l\sin(k\cdot l)+\cos(k\cdot l)-1\right)}{k^2} & -\dfrac{\sin\phi\left(1-\cos(k\cdot l)\right)}{k} & \cos\phi\sin(k\cdot l) \\ \dfrac{\sin\phi\left(k\cdot l\sin(k\cdot l)+\cos(k\cdot l)-1\right)}{k^2} & \dfrac{\cos\phi\left(1-\cos(k\cdot l)\right)}{k} & \sin\phi\sin(k\cdot l) \\ \dfrac{k\cdot l\left(\cos(k\cdot l)-\sin(k\cdot l)\right)}{k^2} & 0 & \cos(k\cdot l) \end{bmatrix} \tag{4-45}$$

$$\mathbf{J}_g(\boldsymbol{u}) = \begin{bmatrix} \dfrac{3\left(l_1l_2+l_1l_3-{l_2}^2-{l_3}^2\right)}{d\tau_{sum}^2\tau_{sqrt}} & -\dfrac{3\left({l_1}^2-l_1l_2-l_2l_3+{l_3}^2\right)}{dl_{sum}^2l_{sqrt}} & -\dfrac{3\left({l_1}^2-l_1l_3-l_2l_3+{l_2}^2\right)}{dl_{sum}^2l_{sqrt}} \\ \dfrac{\sqrt{3}\left(l_3-l_2\right)}{2l_{sqrt}^2} & \dfrac{\sqrt{3}\left(l_1-l_3\right)}{2l_{sqrt}^2} & \dfrac{\sqrt{3}\left(l_2-l_1\right)}{2l_{sqrt}^2} \\ \dfrac{1}{3} & \dfrac{1}{3} & \dfrac{1}{3} \end{bmatrix} \tag{4-46}$$

where $l_{sum} = l_1 + l_2 + l_3$; $l_{sqrt} = \sqrt{{l_1}^2+{l_2}^2+{l_3}^2-l_1l_2-l_1l_3-l_2l_3}$

4.4 Kinematics of the single-tendon-driven case with a rotary motor for rotations

Except for the above mentioned multi-tendon-actuation kinematics, there is another frequently used kinematic model for the continuum manipulator that is actuated with a single tendon but perform the 3D rotation via a rotary motor on the base. In the following, we will briefly discuss this case and derive its corresponding kinematics and Jacobians.

4.4.1 System overview of an exemplary robotic platform

In order to steer a continuum manipulator, an exemplary robotic platform prototype is shown in Figure 4-15. It is a general design in function used for tendon-driven surgical continuum instruments. This platform can be described as an arrangement of three decoupled degrees of freedom (3-DOF): The continuum manipulator is mounted on the upper deck of the robotic platform. Here only a single tendon is routed along the continuum manipulator and driven by a Maxon brushless motor (DC motor EC-max 16, 8W) with the bending knob. It achieves one DOF to bend the flexible manipulator; another rotary motor is used to axially rotate the continuum manipulator mounted deck, which gives the manipulator an extra rotational DOF; the third DOF is to realize the translational shifting of the continuum manipulator via a ball screw.

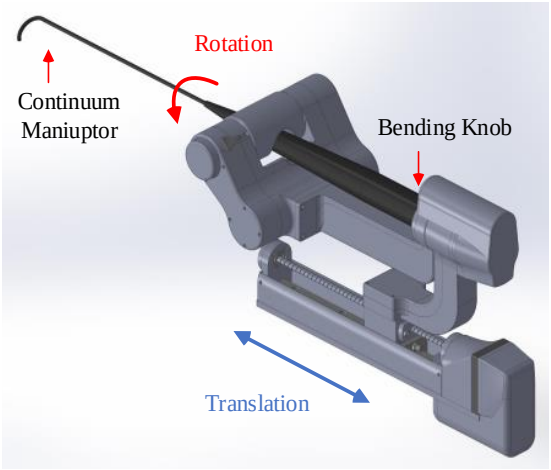


Figure 4-15 3-DOF robotic platform with a continuum manipulator attached.

4.4.2 Derivations of corresponding kinematics and Jacobians

The kinematic model of this discussed continuum manipulator is also derived under the constant-curvature assumption. This validated assumption simplifies modelling and has been widely implemented (Webster & Jones, 2010) to present the kinematics of continuum manipulators. In order to define constant-curvature arcs and locate any points in space, we must define the following coordinate systems in this case, which Figure 4-16 illustrates. First of all, there is a universe coordinate system {*xyz*} established to the initial base plane of the continuum manipulator. It is a non-moving reference system, whose *z*-axis is tangent to the longitudinal axis of the continuum manipulator and its origin is at the base centre. Then, two moving

frames are defined relative to the universe systems: bending plane coordinate system $\{x_b y_b z_b\}$ and distal coordinate system $\{x_d y_d z_d\}$. The bending plane coordinate system is defined such that the manipulator always bends in its $y_b z_b$ plane. In the meanwhile, this frame initially coincides with the reference coordinate system, but with the actuated rotation θ and translation p at the base of the manipulator it moves away from the reference system. The distal coordinate system is attached to the tip of continuum manipulator. Its z_d-axis is tangent to the longitudinal axis of the distal end of the continuum manipulator and $y_d z_d$ plane coincides with the bending plane.

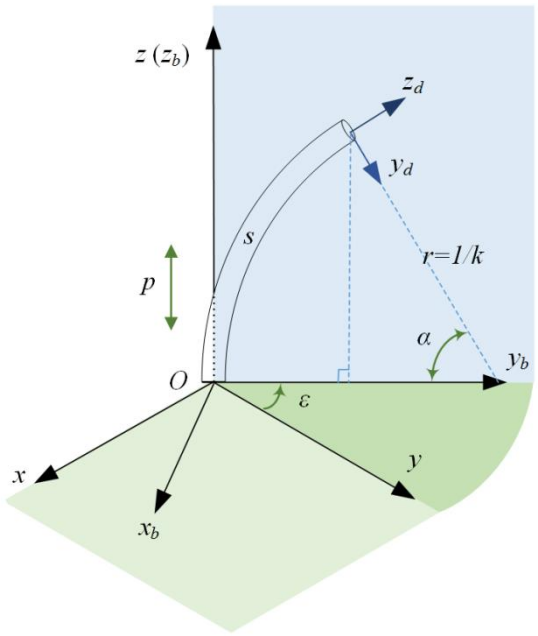


Figure 4-16 Diagram of a continuum manipulator geometry bending in 3D space with defined parameters for kinematic model reference.

4.4.2.1 Kinematics

In this kinematic model, the input actuation space vector is given by $\boldsymbol{\zeta} = [\varphi, \theta, p]^T$, where φ is bending knob input and θ, p describe the handle rotation and translation. Analog to the previous kinematics derivation, in this task space the tip position is also defined by the vector $\boldsymbol{\eta} = [x, y, z]^T$, where x, y, z denote the Cartesian position relative to the universe coordinate system. The completed kinematics model f will map the actuation vector $\boldsymbol{\zeta}$ to the position vector $\boldsymbol{\eta}$, i.e. f: $[\varphi, \theta, p]^T \mapsto [x, y, z]^T$. Throughout this thesis, we omit the time t for all time-variant actuation variables, configuration variables and task space variables.

As discussed in the last section, the kinematic mapping of the continuum manipulator is divided into two submappings. Hence, before we start to derive the kinematics for this continuum manipulator, we also need a bridging vector $\boldsymbol{\chi}$ that describes the manipulator's configuration and divides function f into two parts. The two sub-mappings are same with the previous case: one is robot-specific mapping g that relates actuator space and configuration space while the other is robot-independent mapping h that relates configuration space and task space. We write them as g: $\boldsymbol{\zeta} \mapsto \boldsymbol{\chi}$ and h: $\boldsymbol{\chi} \mapsto \boldsymbol{\eta}$, and we can see that the whole mapping f is the composite mapping of g and h, i.e. $f = h\,(g\,(\cdot))$. Referring back to Figure 4-16, a spatial curve is conventionally parameterized by arc length s, curvature k and orientation ε about the z-axis, where s is constant and k, ε are variables. Besides, because of the translational shifting of the manipulator, the whole body will be seen extended with a rigid straight rod at the proximal end of the bendable arcs. This mathematical quantity in configuration space is equivalent to the translation variable p in actuation space. The actuator variable only appears in this case, while it was not included in our previous kinematic analysis. Therefore, our bridging vector comprises variable terms $\boldsymbol{\chi} = [k, \varepsilon, p]^T$. In our defined coordinate systems, curvature k can only be positive values ($k > 0$) such that the continuum manipulator produces bending toward the $+y$ axis about the $+z$ axis, and orientation ε starts from the $+y$ axis.

In order to derive h, we start by looking at Figure 4-16. As we know, radius $r = 1/k$ and bending angle $\alpha = ks$, and thus we calculate the tip position on the $y_b z_b$ plane to be

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ (1-\cos\alpha)/k \\ \sin\alpha / k \end{bmatrix} \tag{4-47}$$

After that, we rotate the planar model toward anticlockwise direction about the z axis and obtain a spatial model as shown in Figure 4-16. The derivation of spatial position coordinates is obtained by rotating the yz positions about the z-axis using the rotation matrix $\mathbf{R}_z(\varepsilon)$. It is noted that for single tendon steered manipulator the orientation ε of arc parameters is directly equal to the actuated handle rotation θ.

In the next step we consider the displacement p of the prismatic joint. Since the prismatic joint translates the manipulator along the z-axis we simply add p to the z coordinate such that $z = (\sin\alpha)/k + p$. We finally obtain the pose vector $\boldsymbol{x}$ as a function of the bridging parameters k, ε, p, thus completing the h mapping.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{k}\begin{bmatrix} -\sin\varepsilon(1-\cos(ks)) \\ \cos\varepsilon(1-\cos(ks)) \\ \sin(ks) + pk \end{bmatrix} \tag{4-48}$$

Now to find the robot-specific mapping g we derive the bridging parameters as functions of the input actuation variables. As mentioned above, the transformation from actuation space variables θ, p to manipulator shifting and orientation about the z-axis is an identity function which returns the same value that was used as its argument. Then all that's left to do is establish the mapping between bending knob input φ and arc characteristic variable k (or α with $\alpha = ks$). In most practical cases, the bending is nonlinear with respect to the direct knob actuation input due to backlash behaviour of the sheath-guidewire system (Zhang, et al., 2014), (Kesner & Howe, 2011). Therefore, the curve relating the bending knob input and arc characteristic variable exhibits a deadband. By using piecewise linear regression (see Figure 4-17 for example), we empirically estimate the functions between two variables φ and α as

$$\alpha = m_l \varphi + n_l, \quad l = 1, 2, 3 \tag{4-49}$$

where for each actuation interval, m_l, n_l (l=1,2,3) are constant factors and their selection will vary on a case-by-case basis.

We have now found the positions of the manipulator tip as functions of the input actuation variables and known manipulator length s, thus finding the f mapping and completing the kinematic model. We could also use vector notion to write it

$$\boldsymbol{\eta} = f(\boldsymbol{\zeta}) = h(g(\boldsymbol{\zeta})) \tag{4-50}$$

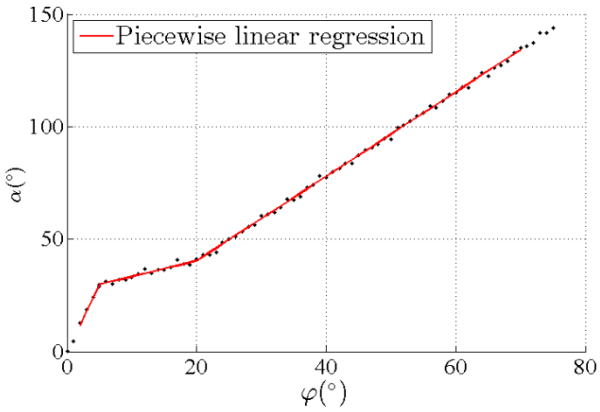


Figure 4-17 The piecewise linear regression curves for modelling the relationship between bending knob input φ and arc characteristic variable α.

4.4.2.2 Jacobians

Given there is the mapping according to (4-50) and taking its partial derivatives, we derive another function – the Jacobians, which maps actuator input velocities to manipulator's tip velocities. For analytical simplicity and given (4-49), we introduce another term $\bar{\zeta}$ to satisfy

$$\dot{\eta} = \mathbf{J}\left(\bar{\zeta}\right)\dot{\bar{\zeta}} \tag{4-51}$$

where $(\dot{})$ is a time derivative, $\mathbf{J} \in \mathbb{R}^{3\times3}$ is the Jacobian matrix and $\bar{\zeta}$ = $[\alpha, \theta, p]^T$. The bending knob control variable φ at last can be derived from the virtual control input α based on (4-49).

It is noted that the whole mapping f is the composite mapping of g and h, thus when taking the partial derivatives of the f we use chain rule to calculate, and we get the explicit Jacobian matrix

$$\mathbf{J} = s\begin{bmatrix} -\sin\theta\left(\frac{\sin\alpha}{\alpha} - \frac{1-\cos\alpha}{\alpha^2}\right) & -\cos\theta(1-\cos\alpha) & 0 \\ \cos\theta\left(\frac{\sin\alpha}{\alpha} - \frac{1-\cos\alpha}{\alpha^2}\right) & -\sin\theta(1-\cos\alpha) & 0 \\ \frac{\cos\alpha}{\alpha} - \frac{\sin\alpha}{\alpha^2} & 0 & \frac{1}{s} \end{bmatrix} \tag{4-52}$$

As we can see, the kinematics and Jacobians in this case are less complexed than that in the previous multi-tendon-driven continuum manipulator case. In Chapter 3, we have mentioned that such simplicity usually benefits the medical applications. Most commercial catheters (Hansen Medical Inc., n.d.) integrate a single tendon, and other flexible instruments including forceps, colonoscopies, endoscopes, etc. are realized in a similar actuation mode for diagnostic and interventional purposes.

4.5 Lagrangian dynamics

This section presents a study on dynamics of the type of our proposed continuum manipulators that possess both bending and contractile capabilities. First, based on curve geometry under the Frenet-Serret frame (Spivak, 1999), kinematics of the continuum manipulator is established. Then, both the kinetic and potential energies are considered for dynamic modelling. By applying the Euler-Lagrangian equation of motion, the system dynamics equation is obtained.

4.5.1 Kinematics under the Frenet-Serret frame

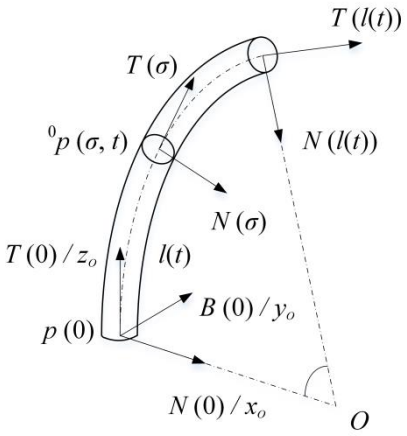


Figure 4-18 Diagram of a continuum manipulator bending geometry in 3D space with coordinate frame illustrated.

In Chapter 2, we discussed the Frenet-Serret frame convention (please refer to 2.4) and the corresponding kinematics of the continuum manipulator has been developed accordingly. The continuum manipulator can be considered with rigid property slice σ of infinitesimal width and being perpendicular along the backbone curve. This methodology has been developed in (Mochiyama & Suzuki, 2002), (Mochiyama & Suzuki, 2003). In order to derive the dynamics of the continuum manipulator, we first define the internal variable $\theta(\sigma,t) \in \mathbb{R}^3$ containing the contraction $l(\sigma,t)$ and curvature $k(\sigma,t)$ at slice σ. The Frenet-Serret frame is illustrated in Figure 4-18. Detailed kinematics equations can be referred to 2.4 in Chapter 2. The above internal variable is written as

$$\theta(\sigma,t) = \begin{bmatrix} l(\sigma,t) \\ k(\sigma,t) \end{bmatrix} \tag{4-53}$$

With the internal variable vector at σ, the extended axis matrix $\bar{\mathbf{A}}(\theta(\sigma,t)) \in \mathbb{R}^{6\times 2}$ is defined as follows. For more details regarding the axis matrix and the extend axis matrix, please refer to (Mochiyama & Suzuki, 2002), (Mochiyama & Suzuki, 2003).

$$\bar{\mathbf{A}}(\theta(\sigma,t)) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T \tag{4-54}$$

In the above development, we assume that the continuum manipulator has no torsion along the backbone. The internal variables are only defined in the bending plane. The conventions utilised in this

section are adopted from (Mochiyama & Suzuki, 2002), (Mochiyama & Suzuki, 2003). Equivalent results are also found in (Tatlicioglu, Walker, & Dawson, 2007), but in different coordinate systems. We expect the work summarized above giving a more clarified illustration to the slice-based methodology for continuum dynamics.

4.5.2 Dynamic modelling

In this part, the dynamic model of a continuum manipulator is developed in the form of the classic Euler-Lagrange representation. Thus, the kinetic and potential energies need to be analysed complying with the curved continuum manipulator.

4.5.2.1 Kinetic Energy

The kinetic energy $K(\sigma,t)$ of slice σ is expressed as follows (Mochiyama & Suzuki, 2002), (Craig, 2005)

$$K(\sigma,t)=\frac{1}{2}m(\sigma)\frac{\partial p_c^T(\sigma,t)}{\partial t}\frac{\partial p_c(\sigma,t)}{\partial t}+\frac{1}{2}\omega(\sigma,t)^T\mathbf{I}(\sigma)\omega(\sigma,t) \tag{4-55}$$

where $\mathbf{I}(\sigma)$ and $m(\sigma)$ are the rotational inertia and translational inertia, respectively, and both of them are time-invariant, thus omitting $t\in\mathbb{R}$ from the expressions. The angular velocity $\omega(\sigma,t)$ at slice σ can be calculated by

$$\omega(\sigma,t)=\int_0^\sigma {}^\sigma\Phi(\eta,t)\frac{\partial a}{\partial t}(\eta,t)\,d\eta \tag{4-56}$$

Then, the kinetic energy $K(\sigma,t)$ can be rewritten as

$$\begin{aligned}K(\sigma,t)&=\frac{1}{2}v^T(\sigma,t)M(\sigma)v(\sigma,t)\\&=\frac{1}{2}\int_0^\sigma\int_0^\sigma\frac{\partial\theta^T(\eta,t)}{\partial t}\bar{A}^T(\eta,t)Ad_{g(\sigma,\eta,t)}^TM(\sigma)Ad_{g(\sigma,\xi,t)}\bar{A}(\xi,t)\frac{\partial\theta(\xi,t)}{\partial t}d\eta d\xi\end{aligned} \tag{4-57}$$

where the adjoint matrix $\mathrm{Ad}_{g(\sigma,\eta,t)}\in\mathbb{R}^{6\times6}$ in terms of rigid body transformation and the inertia matrix $\mathbf{M}(\sigma)\in\mathbb{R}^{6\times6}$ at slice σ can be expressed as follows

$$\mathrm{Ad}_{g(\sigma,\eta,t)}=\begin{bmatrix}{}^\sigma\Phi(\eta,t) & {}^\sigma\hat{p}(\eta,t)-{}^\sigma\hat{p}(\eta,t)\,{}^\sigma\Phi(\eta,t)\\ \mathbf{0}_{3\times3} & {}^\sigma\Phi(\eta,t)\end{bmatrix}$$

$$\mathbf{M}(\sigma)=\begin{bmatrix}m(\sigma)\mathbf{I}_3 & -m(\sigma)\Delta\hat{p}(\sigma)\\ m(\sigma)\Delta\hat{p}(\sigma) & \mathbf{I}(\sigma)\end{bmatrix}$$

where $\Delta p(\sigma)$ means the distance between geometric centre and centre of mass at slice σ, which in our case is zero. $\mathbf{I}_3 \in \mathbb{R}^{3\times3}$ is the identity matrix and $\mathbf{I}(\sigma) \in \mathbb{R}^{3\times3}$, is the inertial tensor of the slice.

$$\mathbf{I}(\sigma) = \frac{mr^2}{4l(t)}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4-58}$$

Thus, the inertial matrix of slice σ can be written as

$$\mathbf{M}(\sigma) = diag\left\{\frac{m}{l(t)}, \frac{m}{l(t)}, \frac{m}{l(t)}, 0, \frac{mr^2}{4l(t)}, 0\right\} \tag{4-59}$$

Therefore, the total kinematic energy of continuum manipulator is calculated as

$$K(t) = \int_0^{l(t)} K(\sigma, t) d\sigma \tag{4-60}$$

4.5.2.2 Potential Energy

In our continuum manipulator, the potential energy consists of three parts, namely gravitational potential energy and elastic potential energy due to bending and contraction.

1. Gravitational potential energy:

The gravitational potential energy of the slice σ can be expressed as

$$P_g(\sigma, t) = -m(\sigma)\, {}^{\sigma}g(\sigma, t) p(\sigma, t) \tag{4-61}$$

where ${}^{\sigma}g(\sigma, t) \in \mathbb{R}^3$ is defined by

$${}^{\sigma}g(\sigma, t) = {}^{\sigma}\Phi^T(0, t)\begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T \tag{4-62}$$

where $g \in \mathbb{R}$ is the gravitational acceleration that is constant. The slice σ gravitational potential energy is illustrated as

$$P_g(\sigma, t) = \frac{mg}{l(t)k(\sigma, t)} \sin\left(\sigma k(\sigma, t)\right) \tag{4-63}$$

Therefore, the total gravitational potential energy of the continuum manipulator is calculated as the integration of the energy in each slice, and it is presented as

$$P_g(t) = \int_0^{l(t)} P_g(\sigma, t) d\sigma \tag{4-64}$$

2. Elastic potential energy due to bending:

According to the contractible continuum manipulator model, the elastic potential energy should be considered in two parts: bending and contraction. The total bending potential energy is calculated as

$$P_b(t) = \frac{1}{2}\int_0^{l(t)} k_b \beta^2(\sigma, t) d\sigma \tag{4-65}$$

where $k_b \in \mathbb{R}$ is the bending spring constant and $\beta(\sigma, t)$ is defined as

$$\beta(\sigma, t) = \pi - \frac{1}{2}\sigma k(\sigma, t) \tag{4-66}$$

3. Elastic potential energy due to contraction:

The elastic potential energy attributed by contraction can be obtained by

$$P_e(t) = \frac{1}{2} k_e \left[d^* - l(t) \right]^2 \tag{4-67}$$

where $k_e \in \mathbb{R}$ is the spring constant associated with contraction and d^* is the relaxed length of manipulator.

Thus, the total potential energy of continuum manipulator is the sum of gravitational potential energy (4-64) and elastic potential energy due to bending (4-65)and contraction (4-67), which is calculated as

$$P(t) = P_g(t) + P_b(t) + P_e(t) \tag{4-68}$$

4.5.2.3 Lagrange Representation

The system Lagranian $L(t)$ is defined as follows

$$L(t) = K(t) - P(t) \tag{4-69}$$

where $K(t)$ and $P(t)$ are total kinetic and potential energies obtained from (4-60) and (4-68) respectively.

The Euler-Lagrangian equation of motion is defined as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau \tag{4-70}$$

where $q(t) \in \mathbb{R}^{2\times 1}$ is the configuration-space joint position variable and defined as follows

$$q(t) = \begin{bmatrix} l(t) & k(t) \end{bmatrix}^T \tag{4-71}$$

Then, the dynamic model of the continuum manipulator can be developed as follows

$$\mathbf{M}(q)\ddot{q} + \mathbf{V}(q, \dot{q})\dot{q} + \mathbf{G}(q) = \tau \tag{4-72}$$

where $\mathbf{M}(q) \in \mathbb{R}^{2\times 2}$ is the manipulator inertia matrix, $\mathbf{V}(q,\dot{q}) \in \mathbb{R}^{2\times 2}$ is the matrix of centrifugal Coriolis torque and $\mathbf{G}(q) \in \mathbb{R}^{2}$ is the matrix about gravitational torques. All entries of the matrices $\mathbf{M}(q)$, $\mathbf{V}(q,\dot{q})$, and $\mathbf{G}(q)$ are defined in Appendix C.

Remark 4-1*:* The manipulator inertia matrix $\mathbf{M}(q)$ and centrifugal Coriolis matrix $\mathbf{V}(q,\dot{q})$ satisfy the following property:

$$\boldsymbol{\xi}^T(\dot{\mathbf{M}} - 2\mathbf{V})\boldsymbol{\xi} = 0, \forall \boldsymbol{\xi} \in \mathbb{R}^3$$

when the matrix $(\dot{\mathbf{M}} - 2\mathbf{V})$ is skew-symmetric.

Chapter 5 Control Strategies for Continuum Manipulators

Continuum manipulators are a rapidly emerging class of robots. However, due to the complexity of their mathematical models and modelling inaccuracies, the development of effective control systems is a particularly challenging task. In this chapter, we investigate the fuzzy logic control methodology and fuzzy-model-based approach, and respectively utilise them to control different tasks relating to continuum manipulators.

In the first part, the fuzzy logic controller is designed for a set-point regulation task of the single-tendon-driven continuum manipulator attached on a rotatory platform; the related kinematics was studies in 4.4. In this case, we firstly derive the state-space model form and linearize it at six operating points. Next, we design a controller for each linearized local model. Last, a fuzzy logic methodology is utilised to smoothly blend the six designed local controllers. The proposed controller is verified in MATLAB simulation and manipulator tip is autonomously navigated to the designated target. Besides, a comparison result for the optimised controller and the non-optimised counterpart is presented.

The second part provides a recent study on kinematic control of continuum manipulators using a fuzzy-model-based approach. The objective is to enable the end-effector of the manipulator to accomplish autonomous execution of command tracking. The proposed fuzzy-model-based tracking control system consists of a fuzzy model representing the continuum manipulator, a fuzzy controller and reference models given by two different trajectory tracking tasks. This control methodology is developed to guarantee system stability and facilitate the chosen approach of control synthesis. The superior performance of this controller is tested and validated in MATLAB simulation and further analysed by comparing to classical controllers found in the literature. The experiments on a rapid-prototyped continuum manipulator further verify the feasibility and the advantages of this fuzzy controller in the presence of modelling discrepancies and hardware defects.

5.1 Introduction

Continuum manipulators are mainly characterised by their ability to continuously bend along the length of their structure; further, due to the inherent compliance, these manipulators demonstrate appealing flexibility and allow safe interactions in constrained environments. Although continuum robotics is still in its infancy, considerable current research is focusing on the development of both hardware and machine learning methods for such continuum robots, including design, modelling, control, and learning (Walker, 2013). Continuum robot manipulators have made inroads into a rapidly growing number of applications across different sectors, ranging from industrial operations (Hirose, 1993) to health care and domestic environments (Simaan, et al., 2009). Compared with conventional manipulators with segmented rigid-links, the architecture concept and actuation principles for continuum manipulators are fundamentally different – they often mimic biological trunk or tentacle behaviours and manipulate objects in ways similar to how the biological role models do it. Particularly, continuum manipulators emphasise "whole arm manipulation" of a wide range of objects (Salisbury, 1987), which is even performed without prior knowledge of the shape of the object. Detailed surveys of the state-of-the-art and continuum manipulator designs are given in (Walker, 2013), (Webster & Jones, 2010). Frequently applied continuum manipulator structures are tendon-driven flexible backbone designs (Kim, Cheng, Kim, & Iagnemma, 2013), pneumatically actuated bellow-integrated designs (Festo Corporate, 2010), concentric tube designs (Webster, Joseph, & Cowan, 2009), and soft body structures with locally actuated cells (Cianchetti, et al., 2013).

With regard to robotic control for continuum manipulators, researchers have proposed real-time feedback control to improve the system performance during autonomous execution of command tracking tasks (Penning, Jung, Borgstadt, Ferrier, & Zinn, 2011), (Kesner & Howe, 2011). A variety of sensing techniques like electromagnetic tracking and image/video (Koolwal, Barbagli, Carlson, & Liang, 2010) are used to provide manipulator's real-time pose, shape or other information, and based on that, closed-loop control strategies are investigated. On the other hand, researchers have done a lot of research in kinematic/dynamic modeling in order to predict continuum structure behavior (Camarillo, Milne, Carlson, Zinn, & Salisbury, 2008). From the perspective of control theory, this model-based prediction acts as feedforward component of a closed loop. And the more precise the prediction of the manipulator's behavior is, the better the feedback control performance will be. However, in contrast, considering that continuum manipulator articulates due to inherent compliance and conforms based on unknown constraints and obstacles the authors in (Yip & Camarillo, 2014) firstly utilize a task-space closed-loop

controller that is only based on empirical estimates of the real-time Jacobian but without using a model. Particularly, fuzzy control algorithm has been investigated and successfully applied to motion control of the traditional rigid-link manipulators with different specifications (Althoefer, 1997), (Althoefer & Seneviratne, 1999) and this inspired the following research to adopt the fuzzy control approaches to continuum manipulators.

In the first part of this chapter, we will investigate the feasibility of fuzzy logic control methodology (Althoefer, Krekelberg, Husmeier, & Seneviratne, 2001), (Althoefer, Seneviratne, Zavlangas, & Krekelberg, 1998) that combines multiple state-feedback controllers to regulate the tip position of a continuum manipulator with the single-tendon-driven and rotatory platform. The kinematics for the continuum manipulator in this case is was discussed in 4.4, which refers to the one with the single-tendon-driven and rotary platform. Then, the model is converted to state-space model form. We carry out an off-line linearization of the state-space represented model at several tip positions. And then we design specific linear controller for each local model. At last, fuzzy logic methodology is employed to smoothly blend those designed local controllers to obtain a nonlinear controller. Simulation results and analysis are reported subsequently. A conclusion and plan for set-point regulation tasks is given in the end of this part.

In the second part, we present a fuzzy-model-based approach for controlling a continuum manipulator. The controller was designed based on stability analysis for general continuous-time nonlinear systems (Lam & Li, 2013). First, based on the kinematic model derived in 4.3, we analyse its successive state-space model. Then, a fuzzy model is established to represent this state-space model by using a local approximation technique (Tanaka & Wang, 2004). We design the fuzzy controller based on the stability conditions proposed in (Lam & Li, 2013). This controller enables the states of our continuum manipulator to track a desired reference model. The fuzzy-model-based approach can suppress the tracking error according to H_∞ performance based on the Lyapunov stability theory. Compared with open-loop feedforward control which is highly dependent on the accuracy of the model in real-time control, our closed-loop control is adequate to accommodate the online trajectory adjustments and has effective trajectory tracking capabilities. Although there commonly exists a certain modelling error between the established fuzzy model and the physical nonlinear model, the stability and performance of the specified tracking task can still be accomplished. Compared with other (pseudo-)inverse Jacobian based kinematic control systems, the proposed method does not require online updating of the Jacobian, nor rely on continuously updated estimations of the Jacobian. It responses to sensor inputs, thus also providing a closed-form low-computation solution of a motion control problem with respect to continuum

manipulators. To the best of our knowledge, this is the first work of achieving task space closed-loop control proposed with respect to a continuum manipulator using a fuzzy-model-based approach.

5.2 Fuzzy logic control syntheses for set-point regulation tasks

The objective of the control task is to navigate the manipulator tip to reach designated targets by actuating the robotic apparatus. Figure 5-1 illustrates the control architecture. In view of the model complexities and high computational cost of using inverse Jacobians, we linearize the system model to simplify the control design and consider state-space methods to derive local linear controllers. After that, in order to reshape the actual system, we use fuzzy logic to create a nonlinear controller overall.

In 4.4, we have given a common model of the discussed continuum manipulator and its corresponding Jacobian matrix which appear to be nonlinear. Owing to the nonlinear dynamic of the continuum manipulator, the controller design is difficult. In view of this, we linearize the state-space system at a number of different operating points and design a specific optimal controller for each of the system linearizations. A fuzzy logic controller (Lee, Lam, Leung, & Tam, 2003), (Lee, Lam, Leung, & Tam, 2001) is then proposed to smoothly blend the above designed local controllers. Details about control synthesis are shown in the following.

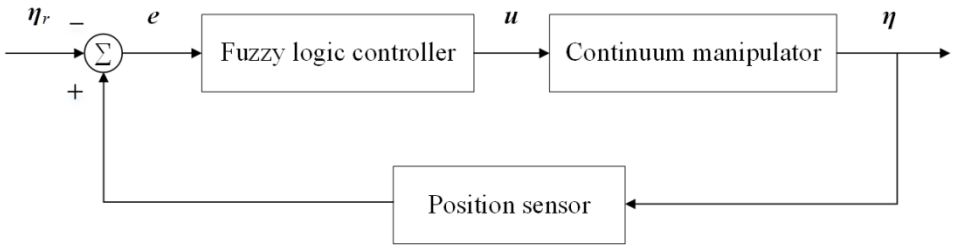


Figure 5-1 Closed-loop control architecture. A fuzzy logic controller is designed based on state-feedback controller.

5.2.1 State-space representation

Referring back to (4-51) that gives the system model and given (4-52), we aim to obtain the variables α, θ correspondingly as a function of x, y, z, in order to derive the state-space representation.

Referring back to the 2D bending plane of Figure 4-16, we derive the following relations via geometric method.

$$\cos\alpha = \frac{r-\sqrt{x^2+y^2}}{r}; \quad \theta = \cos^{-1}\left(\frac{y}{r(1-\cos\alpha)}\right) \tag{5-1}$$

Obtaining the numerical solution of the variables α, θ in (5-1) with radius $r = s/\alpha$, we then substitute them into the system model to have the state-space model written in the following form

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(x,y)\dot{\bar{\boldsymbol{\zeta}}} \tag{5-2}$$

where $\bar{\boldsymbol{\zeta}} = [\alpha, \theta, p]^T$, denoting the actuator-space variable and it was defined in (4-51).

The constant target position in continuum manipulator tip workspace is defined as $\left[x_r, y_r, z_r\right]^T$. And we define the error as $e_x = x - x_r$, $e_y = y - y_r$, $e_z = z - z_r$. Substituting them into (5-2) and simplifying the equation, we have

$$\dot{\boldsymbol{e}} = \mathbf{J}\left(e_x + x_r, e_y + y_r\right)\boldsymbol{u} \tag{5-3}$$

where $\boldsymbol{e} = \left[e_x, e_y, e_z\right]^T$ and $\boldsymbol{u} = \left[\dot{\alpha}, \dot{\theta}, \dot{p}\right]^T$.

5.2.2 Linearization and local controller design

In order to linearize the above nonlinear state-space model, we first define the workspace of our study case continuum manipulator. The constant length s is assumed to be 1. The range of x is [–0.5, –0.01], y [–0.5, 0.5], and z [0, 2]. Then, we choose the values $[e_x+x_r, e_y+y_r]^T$ of six different operating points in the xy plane $[-0.01, -0.5]^T$, $[-0.01, 0]^T$, $[-0.01, 0.5]^T$, $[-0.5, -0.5]^T$, $[-0.5, 0]^T$, $[-0.5, 0.5]^T$ within the defined workspace and substitute them into (5-3), from which we obtain six specific Jacobian matrices

$$\mathbf{J}_1 = \begin{bmatrix} -0.0071 & 0.5 & 0 \\ -0.3563 & -0.01 & 0 \\ -0.3262 & 0 & 1 \end{bmatrix} \quad \mathbf{J}_2 = \begin{bmatrix} -0.4999 & 0 & 0 \\ 0 & -0.01 & 0 \\ -0.0067 & 0 & 1 \end{bmatrix}$$

$$\mathbf{J}_3 = \begin{bmatrix} -0.0071 & -0.5 & 0 \\ 0.3563 & -0.01 & 0 \\ -0.3262 & 0 & 1 \end{bmatrix} \quad \mathbf{J}_4 = \begin{bmatrix} -0.0732 & 0.5 & 0 \\ -0.0732 & -0.5 & 0 \\ -0.4352 & 0 & 1 \end{bmatrix}$$

$$\mathbf{J}_5 = \begin{bmatrix} -0.3564 & 0 & 0 \\ 0 & -0.5 & 0 \\ -0.3262 & 0 & 1 \end{bmatrix} \quad \mathbf{J}_6 = \begin{bmatrix} -0.0732 & -0.5 & 0 \\ 0.0732 & -0.5 & 0 \\ -0.4352 & 0 & 1 \end{bmatrix}$$

With respect to each above calculated Jacobian matrix, we design a local linear controller in the form of

$$\boldsymbol{u}_i = \mathbf{G}_i \boldsymbol{e}, \quad i = 1, 2, \ldots, 6 \tag{5-4}$$

where $\mathbf{G}_i$ is the controller gain to be designed in the framework of a linear control strategy scheme.

Furthermore, to optimise the trajectory of tip navigation and reduce the control energy we use linear-quadratic regulator (LQR) method to design the state-feedback controller such that the following quadratic cost function is minimized.

$$K_i = \int_0^\infty \left(\boldsymbol{e}^T \mathbf{Q} \boldsymbol{e} + \boldsymbol{u}_i^T \mathbf{R} \boldsymbol{u}_i \right) dt, \quad i = 1, 2, \ldots, 6 \tag{5-5}$$

where **Q** and **R** are weighting matrices.

By choosing weighting matrix **Q** = **I** and weighting matrix **R** = **I**, where **I** is identity matrix, we get

$$\mathbf{G}_1 = \begin{bmatrix} 0.0194 & 0.9721 & 0.2339 \\ -0.9998 & 0.0200 & 0 \\ 0.0047 & 0.2338 & -0.9723 \end{bmatrix} \quad \mathbf{G}_2 = \begin{bmatrix} 1 & 0 & 0.0044 \\ 0 & 1 & 0 \\ 0.0044 & 0 & -1 \end{bmatrix}$$

$$\mathbf{G}_3 = \begin{bmatrix} 0.0194 & -0.9721 & 0.2339 \\ 0.9998 & 0.0200 & 0 \\ 0.0047 & -0.2338 & -0.9723 \end{bmatrix} \quad \mathbf{G}_4 = \begin{bmatrix} 0.6578 & 0.6578 & 0.3669 \\ -0.7071 & 0.7071 & 0 \\ 0.2594 & 0.2594 & -0.9303 \end{bmatrix}$$

$$\mathbf{G}_5 = \begin{bmatrix} 0.9723 & 0 & 0.2338 \\ 0 & 1 & 0 \\ 0.2338 & 0 & -0.9723 \end{bmatrix} \quad \mathbf{G}_6 = \begin{bmatrix} 0.6578 & -0.6578 & 0.3669 \\ 0.7071 & 0.7071 & 0 \\ 0.2594 & -0.2594 & -0.9303 \end{bmatrix}$$

For comparison purpose, we also use the pole placement method by setting the eigenvalues of the linearized closed-loop system to be −0.1, −0.2, −0.3 and correspondingly get the controller gains. The choice of these closed loop poles ensure that the system is stabilized and there is no oscillation. The results will be shown in the subsequent simulations.

5.2.3 Fuzzy logic control

A fuzzy logic controller having six rules is proposed to combine the above designed six local linear controllers. The membership functions are defined as

$$m_{11}(e_x + x_r) = 1 - \frac{1}{1 + \exp\left(-\dfrac{(e_x + x_r)^2 - 0.125}{0.02} \right)} \tag{5-6}$$

$$m_{12}(e_x + x_r) = 1 - m_{11}(e_x + x_r) \tag{5-7}$$

$$m_{21}(e_y + y_r) = 1 - \frac{1}{1 + \exp\left(-\dfrac{(e_y + y_r) + 0.2}{0.05} \right)} \tag{5-8}$$

$$m_{23}(e_y + y_r) = \frac{1}{1 + \exp\left(-\dfrac{(e_y + y_r) - 0.2}{0.05} \right)} \tag{5-9}$$

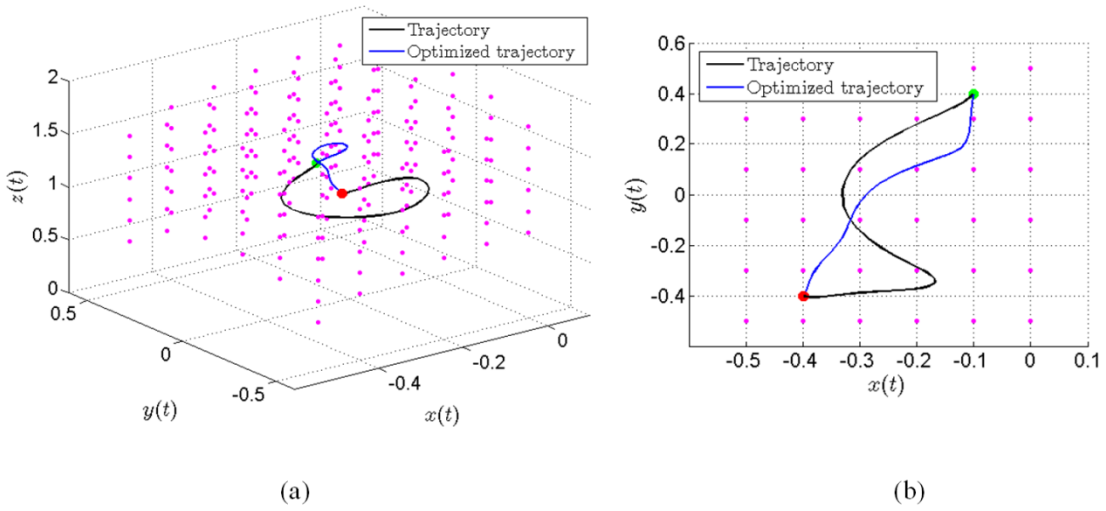
$$m_{22}(e_y + y_r) = 1 - m_{21}(e_y + y_r) - m_{23}(e_y + y_r) \tag{5-10}$$

The grades of memberships, w_1 to w_6, of rule 1 to rule 6, which are respectively for local controllers $\boldsymbol{u}_1$ to $\boldsymbol{u}_6$, are defined as follows: $w_1 = m_{11} \times m_{21}$; $w_2 = m_{11} \times m_{22}$; $w_3 = m_{11} \times m_{23}$; $w_4 = m_{12} \times m_{21}$; $w_5 = m_{12} \times m_{22}$; $w_6 = m_{12} \times m_{23}$. The output of the fuzzy logic controller is given by

$$\boldsymbol{u} = \frac{\sum_{i=1}^{6} w_i \boldsymbol{u}_i}{\sum_{i=1}^{6} w_i} \tag{5-11}$$

5.3 Set-point regulation simulation results and analysis

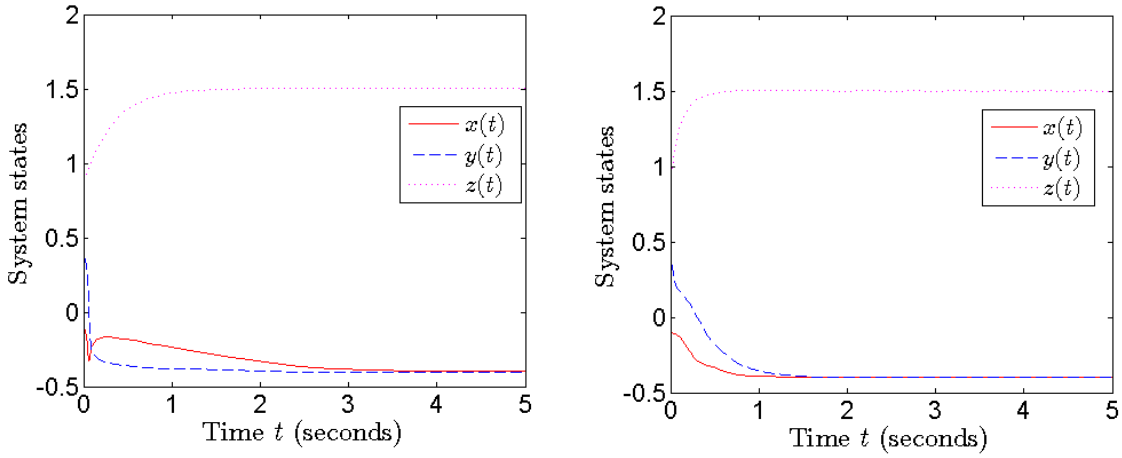
We implement the proposed fuzzy logic controller in MATLAB simulation to investigate its performance. The simulation environment contains an aforementioned tip workspace of a single-tendon-driven continuum manipulator. The manipulator's mathematical model is utilised and `ode` function command in MATLAB is executed to generate the continuous navigation path from the defined initial position $[-0.1, 0.4, 0.8759]^T$ to target reference site $[-0.4, -0.4, 1.5]^T$. The trajectories of the continuum manipulator tip in 3D workspace are captured and plotted in Figure 5-2(a). The controllers associated with two types of different feedback gain derivation techniques generate two trajectories in workspace. We can see that both trajectories have achieved the regulation task with corresponding fuzzy logic controller. However, in Figure 5-2(b), the 2D view of the two trajectories obviously reveals the superiority of the optimised controller that the blue line corresponding to the optimised trajectory is smoother and shorter than the black line.



(a) (b)

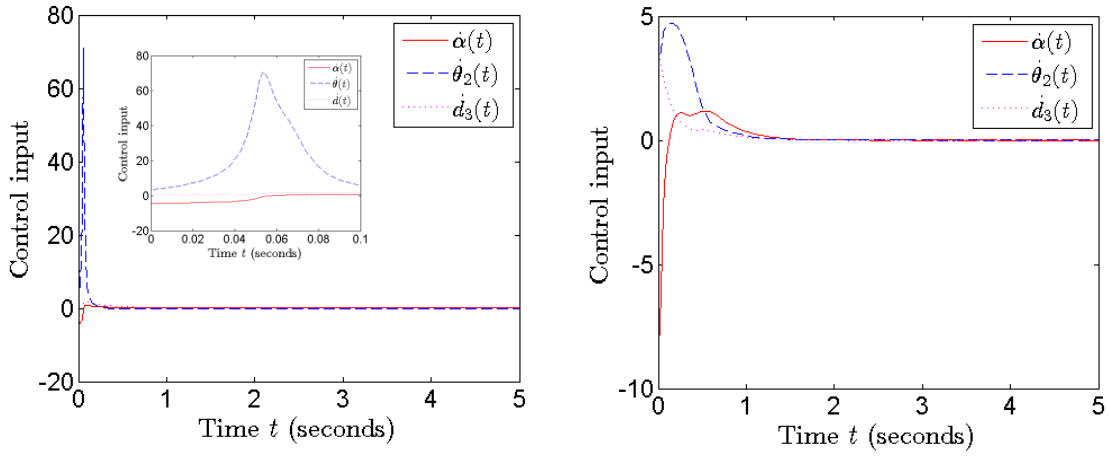
Figure 5-2 Trajectory of the continuum manipulator tip controlled by the proposed fuzzy logic controller. A green dot indicates the initialized position; a red dot indicates the reference position; pink dots indicate the tip workspace. (a) 3D screenshot of the workspace and the tip trajectory of the continuum manipulator; (b) 2D *xy* plane screenshot.

Figure 5-3 further gives one group of comparison for optimised fuzzy state-feedback controller and non-optimised counterpart. Figure 5-3(a) and Figure 5-3(b) take the state-space variables *x*, *y*, *z* into consideration, where the changing curves of three coordinate components with respect to each controller are plotted. We can see that for both cases each component reaches its reference in the end, while the curve of component *x* in Figure 5-3(a) shows unwanted vibrations at the very beginning. With the optimised controller the approaches the steady states more smoothly and quickly. In this regard, the optimised controller is better than the former in terms of physical stability performance.



(a) (b)

Figure 5-3 (a) The changing curve of coordinate components in state space with non-optimised controller; (b) The changing curve of coordinate components in state space with optimised controller.



(a) (b)

Figure 5-4 (a) Control input with regard to time during the regulation task with non-optimised controller. (b) Control input with regard to time during the regulation task with optimised controller.

Figure 5-4(a) and Figure 5-4(b) give another group of comparison regarding the control effort. The plotted curves indicate that the required control inputs especially for $\dot{\theta}$ (blue dashed line) with optimised controller are smaller than that with non-optimised controller. That means it is much easier to control the system with the optimised controller.

5.4 Conclusion and future works for set-point regulation tasks

A fuzzy logic controller has been proposed to control an exemplary continuum manipulator with the single-tendon-driven which is frequently used in surgical application. Firstly, we utilised the kinematics and Jacobians of the single-tendon-driven continuum manipulator, and gave the state-space presentation of the system. Then, in order to control the nonlinear system, we linearized the state-space model at six different operating points and design a linear controller for each of the local models. Finally, a fuzzy logic methodology was utilised to obtain a nonlinear controller that smoothly blends the six designed local controllers. Moreover, in the process of designing the six controllers we also used two different methods – linear-quadratic regulator (LQR) method and pole placement method – to obtain the feedback gain, thus getting an optimised controller. We tested the proposed fuzzy logic controller in MATLAB simulations and the manipulator tip was successfully navigated to the designated targets. A comparison result for optimised and non-optimised controllers was presented, which reveals the superiority of the former.

The related next-step work can further refine the model of continuum manipulator and take the practical influencing factors, such as friction and environmental disturbance, into consideration (Althoefer & Bugmann, 1995), (Althofer & Fraser, 1996). Besides, extending the implementation to multi-tendon-driven more flexible continuum manipulator case is promising. Finally, we conduct practicality experiments and expect to implement proposed control strategies into clinical test with hospital collaborations.

5.5 Fuzzy-model-based approach for trajectory tracking tasks

The objective of this kinematic control task is to find a solution with respect to the actuation space variables to enable the end-effector of the continuum manipulator to track a desired trajectory. Figure 5-5 illustrates the control architecture. Afterwards, we first introduce the design procedures step-by-step according to the literature (Lam & Leung, 2011). Then the state-space model with respect to the continuum manipulator is proposed, based on which the fuzzy model can be developed. We specify two

different trajectory tracking tasks and accordingly design two sets of feedback gains in the fuzzy controllers. Details about control synthesis are shown in the following subsections.

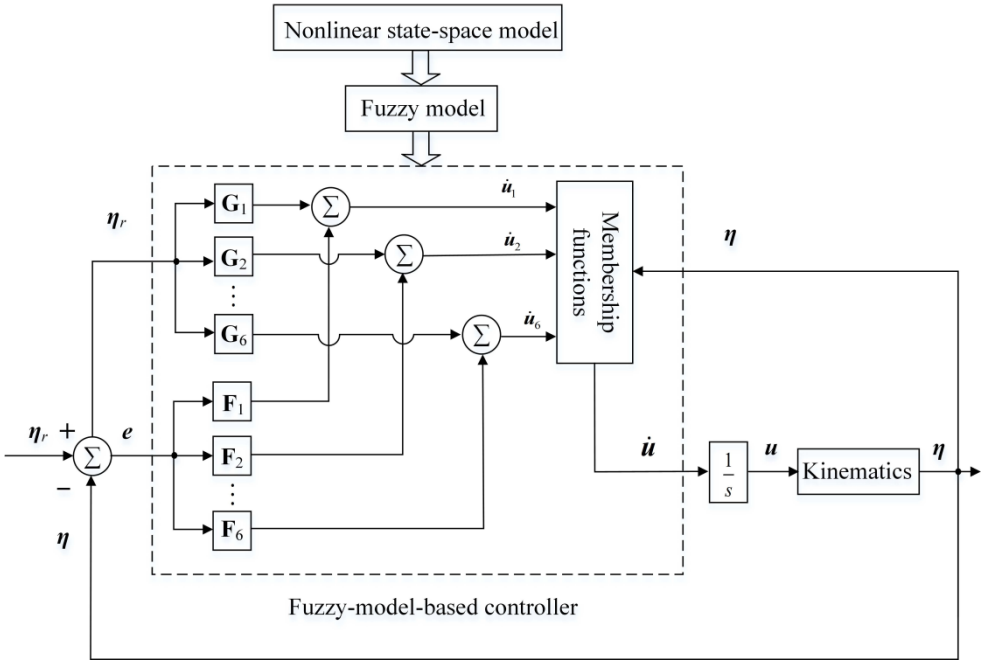


Figure 5-5 Overview of a task space closed-loop tracking control system using the fuzzy-model-based approach for continuum manipulators. $\boldsymbol{\eta}_r$ represents the desired end-effector trajectory in task space. $\mathbf{G}_j$ and $\mathbf{F}_j$, (j = 1, 2, ⋯, 6), are feedback gains. ($\dot{}$) is a time derivative and $\dot{\boldsymbol{u}}$ denotes the end-effector motion velocity. The feedback information is acquired with position sensors.

5.5.1 Polynomial fuzzy-model-based stability conditions

The methodology of the fuzzy-model-based control is summarised in this subsection based on (Lam & Li, 2013).

5.5.1.1 Polynomial fuzzy model

In order to apply the fuzzy-model-based stability analysis, the polynomial fuzzy model is employed to represent the system state model of the continuum manipulator. The polynomial fuzzy model is constructed by using membership functions to blend the local polynomial models. The p-rule polynomial fuzzy model describing the behaviour of a general nonlinear model can be defined as (Tanaka, Yoshida, Ohtake, & Wang, 2009)

$$\dot{\boldsymbol{x}} = \sum_{i=1}^{p} w_i(\boldsymbol{y})\left(\mathbf{A}_i(\boldsymbol{x})\bar{\boldsymbol{x}}(\boldsymbol{x}) + \mathbf{B}_i(\boldsymbol{x})\boldsymbol{v}\right) \tag{5-12}$$

$$\boldsymbol{y} = \mathbf{C}\bar{\boldsymbol{x}}(\boldsymbol{x}) \tag{5-13}$$

where $\boldsymbol{x}$ denotes the system state vector; $\boldsymbol{y}$ denotes the output vector; $w_i(\boldsymbol{y})$ is the normalised grade of membership; $\mathbf{A}_i(\boldsymbol{x})$ and $\mathbf{B}_i(\boldsymbol{x})$ are the known polynomial system and input matrices, respectively; $\bar{\boldsymbol{x}}(\boldsymbol{x})$ is a vector of monomials in $\boldsymbol{x}$; $\boldsymbol{v}$ is the input vector; $\mathbf{C}$ is a constant output matrix. It is assumed that $\bar{\boldsymbol{x}}(\boldsymbol{x}) = \boldsymbol{0}$ iff $\boldsymbol{x} = \boldsymbol{0}$.

5.5.1.2 Reference model

The reference model mathematically describes the desired trajectory. It is specified by users and later is utilised in the fuzzy-model-based stability analysis for the tracking control of a continuum manipulator. The reference model is defined as follows (Lam & Li, 2013):

$$\dot{\boldsymbol{x}}_r = \mathbf{A}_r\bar{\boldsymbol{x}}_r(\boldsymbol{x}_r) + \mathbf{B}_r\boldsymbol{r} \tag{5-14}$$

$$\boldsymbol{y}_r = \mathbf{C}\bar{\boldsymbol{x}}_r(\boldsymbol{x}_r) \tag{5-15}$$

where $\boldsymbol{x}_r$ denotes the state vector of the reference model; $\mathbf{A}_r$ and $\mathbf{B}_r$ are the constant system and input matrices, respectively; $\bar{\boldsymbol{x}}_r(\boldsymbol{x}_r)$ is a vector with monomials in $\boldsymbol{x}_r$ as the entries; $\boldsymbol{r}$ denotes the reference input vector; $\boldsymbol{y}_r$ denotes the output vector of the reference model.

5.5.1.3 Output-feedback polynomial fuzzy controller

The basic idea of trajectory tracking is to continuously reduce the discrepancies between the desired position and the actual position. A polynomial fuzzy controller is employed here to track the trajectory without the online computation of a (pseudo-)inverse Jacobian matrix. This fuzzy controller is designed based on the concept of the parallel distributed compensation (PDC) (Wang, Tanaka, & Griffin, 1996). In other words, the membership functions integrated in the fuzzy controller are the same as those in (5-12). The output-feedback polynomial fuzzy controller is defined as follows (Lam & Li, 2013):

$$\boldsymbol{v} = \sum_{j=1}^{p} w_j(\boldsymbol{y})\left(\mathbf{F}_j(\boldsymbol{h})\mathbf{C}\bar{\boldsymbol{e}} + \mathbf{G}_j(\boldsymbol{h})\mathbf{C}\bar{\boldsymbol{x}}_r(\boldsymbol{x}_r)\right) \tag{5-16}$$

where we define $\boldsymbol{h} = [\boldsymbol{y}^T, \boldsymbol{y}_r^T]^T$; $\mathbf{F}_j(\boldsymbol{h})$ and $\mathbf{G}_j(\boldsymbol{h})$ are the polynomial feedback gains to be determined; the tracking error is defined by $\bar{\boldsymbol{e}} = \bar{\boldsymbol{x}}(\boldsymbol{x}) - \bar{\boldsymbol{x}}_r(\boldsymbol{x}_r)$.

5.5.1.4 H_∞ performance of tracking control

The tracking performance can be governed by an H_∞ performance index which can be adjusted by the user to minimise the tracking error $\bar{\boldsymbol{e}}$ in (5-16). It origins from the Lyapunov-based stability analysis. The H_∞ performance of tracking control is defined as follows (Lam & Li, 2013):

$$\frac{\int_0^{t_f} \mathbf{z}_1^T \mathbf{z}_1 dt - V(0)}{\int_0^{t_f} \left(\sigma_1^2 \mathbf{z}_2^T \mathbf{z}_2 + \sigma_2^2 \mathbf{z}_3^T \mathbf{z}_3\right) dt} \leq 1 \tag{5-17}$$

where t_f is the termination time of tracking control; σ_1 and σ_2 are the pre-defined scalars; $\boldsymbol{z}_1 = \mathbf{X}(\tilde{\boldsymbol{x}})\boldsymbol{\Gamma}^{-1}\bar{\boldsymbol{e}}$, $\boldsymbol{z}_2 = \mathbf{X}(\tilde{\boldsymbol{x}})^{-1}\boldsymbol{\Gamma}^{-1}\boldsymbol{x}_r$, $\boldsymbol{z}_3 = \boldsymbol{r}$; $\mathbf{X}(\tilde{\boldsymbol{x}}) = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_{22}(\tilde{\boldsymbol{x}}) \end{bmatrix}$ is termed as a symmetric decision variable which can be obtained in MATLAB; $\boldsymbol{\Gamma} = \left[\mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1} \quad ortc(\mathbf{C}^T)\right]$ and $ortc(\cdot)$ denotes the orthogonal complement; $V(t) = \boldsymbol{\rho}^T \mathbf{X}(\tilde{\boldsymbol{x}})^{-1}\boldsymbol{\rho}$ is the polynomial Lyapunov function candidate, $\boldsymbol{\rho} = \boldsymbol{\Gamma}^{-1}\bar{\boldsymbol{e}}$; $\tilde{\boldsymbol{x}} = [x_{j_1}, x_{j_2}, ..., x_{j_q}, x_{r_{k_1}}, x_{r_{k_2}}, ..., x_{r_{k_s}}]^T$, the subscripts j_1, j_2, …, j_q, are the row indices that the entries of the entire row of $\mathbf{B}_i(\boldsymbol{x})$ for all i are all zeros, the subscripts k_1, k_2, …, k_s, are the row indices that the entries of the entire row of $\mathbf{B}_r$ are all zeros.

5.5.1.5 Stability conditions of the polynomial fuzzy-model-based control systems

The defining feature and also the superiority of the fuzzy-model-based approach is that various control problems such as trajectory tracking and H_∞ performance can be systematically analysed whilst ensuring the system stability. This gives the theoretical support to physically implement the designed controller. It is derived based on the Lyapunov stability theory.

Before proceeding further, we first describe the following notations which will be employed in the theorem. A polynomial $p(\boldsymbol{x})$ is a sum of squares (SOS) if it can be written as $p(\boldsymbol{x}) = \sum_{j=1}^{m} q_j(\boldsymbol{x})^2$ where $q_j(\boldsymbol{x})$ is polynomial and m is a non-zero integer. Thus, if the condition "$p(\boldsymbol{x})$ is an SOS" holds, then we have $p(\boldsymbol{x}) \geq 0$. SOSTOOLS is a third-party MATLAB toolbox to numerically find solutions to SOS conditions (Prajna, Papachristodoulou, & Parrilo, 2002).

Theorem 5-1 (Lam & Li, 2013)

The designed polynomial fuzzy controller in (5-16) is guaranteed to enable the states of the polynomial fuzzy model in (5-12) representing the physical nonlinear system to track a desired reference model in

(5-14) subject to an H_∞ performance of (5-17) if there exists decision variables $\mathbf{X}(\tilde{\boldsymbol{x}})$ referring to (5-17), $\mathbf{M}_j(\boldsymbol{h})$ and $\mathbf{N}_j(\boldsymbol{h})$, ($j$ = 1, 2, …, p), such that the following SOS conditions are satisfied.

$$\boldsymbol{v}_1^T \left(\mathbf{X}(\tilde{\boldsymbol{x}}) - \varepsilon_1(\tilde{\boldsymbol{x}})\mathbf{I}\right)\boldsymbol{v}_1 \text{ is SOS;}$$

$$-\boldsymbol{v}_2^T \left(\Xi_{ij}(\boldsymbol{x}, \boldsymbol{x}_r) + \Xi_{ji}(\boldsymbol{x}, \boldsymbol{x}_r) + \varepsilon_2(\boldsymbol{x}, \boldsymbol{x}_r)\mathbf{I}\right)\boldsymbol{v}_2 \text{ is SOS } \forall j = 1, 2, ..., p; i < j \text{ ;}$$

where $\mathbf{v}_1$ and $\mathbf{v}_2$ are arbitrary vectors independent of $\boldsymbol{x}$ and $\boldsymbol{x}_r$; $\varepsilon_1(\tilde{\boldsymbol{x}})$ and $\varepsilon_2(\boldsymbol{x}, \boldsymbol{x}_r)$ are pre-defined positive polynomials; the technical details of $\Xi_{ij}(\boldsymbol{x}, \boldsymbol{x}_r)$can be found from Equation (29) in (Lam & Li, 2013). The feedback gains can be obtained by

$$\mathbf{F}_j(\boldsymbol{h}) = \mathbf{M}_j(\boldsymbol{h})\mathbf{X}_{11}^{-1}, \qquad \mathbf{G}_j(\boldsymbol{h}) = \mathbf{N}_j(\boldsymbol{h})\mathbf{X}_{11}^{-1}$$

5.5.2 State-space representation

The derived Jacobian in (4-44) reveals the velocity-level kinematics and fully describes the continuous-time dynamic system. From the control aspect, the mathematical description of the system is expressed as

$$\begin{aligned} &\dot{\boldsymbol{\eta}} = \mathbf{J}\left(g^{-1}\left(h^{-1}(\boldsymbol{\eta})\right)\right)\dot{\boldsymbol{u}} \\ &\Leftrightarrow \dot{\boldsymbol{\eta}} = \mathbf{J}\left(\boldsymbol{\eta}\right)\dot{\boldsymbol{u}} \end{aligned} \tag{5-18}$$

where the equation is known as the state-space model and $\boldsymbol{u} = g^{-1}\left(h^{-1}(\boldsymbol{\eta})\right)$ can be obtained analytically by solving their respective parts of forward kinematics.

With the substitution of the above state-space model in the time domain, the state-space controller design techniques such as (Lam & Li, 2013), (Tanaka & Wang, 2004), are enabled towards a dynamic system for the continuum manipulator.

5.5.3 Fuzzy model construction via the local approximation

In order to represent the continuum manipulator state-space model embodied in (5-18) by a fuzzy model, a local approximation technique is utilised. In our case, the task space in Cartesian coordinate system for a continuum manipulator with 0.01m diameter (i.e. d = 0.005m in (4-39), (4-40), (4-41)) is specified as

$$\mathbb{D}^3 = \left\{x, y, z \mid x \in [0.015, 0.075], y \in [-0.075, 0.075], z \in [0.015, 0.15]\right\} \text{ (Unit: m)} \tag{5-19}$$

Based on this range of interest, we approximate the state-space model at six different local sets of system states, i.e. $\boldsymbol{\eta}_1$ = [0.015, –0.075, 0.075]^T, $\boldsymbol{\eta}_2$ = [0.015, 0, 0.075]^T, $\boldsymbol{\eta}_3$ = [0.015, 0.075, 0.075]^T, $\boldsymbol{\eta}_4$ = [0.075, –0.075, 0.075]^T, $\boldsymbol{\eta}_5$ = [0.075, 0, 0.075]^T, $\boldsymbol{\eta}_6$ = [0.075, 0.075, 0.075]^T. Note that more local sets of

system states can be used to establish a more accurate fuzzy model. However, it will lead to higher computational demand. Other advanced fuzzy modelling techniques can be employed to find a better trade-off between the accuracy and computational burden. In this paper, the system state *z* is only approximated at one point to lower the computational demand. Then, the local state-space models with respect to each set of system states can be obtained like:

$$\mathbf{A}_i = \mathbf{0}, \qquad \mathbf{B}_i = \mathbf{J}(\boldsymbol{\eta}_i), \qquad i = 1, 2, ..., 6, \tag{5-20}$$

where the derived input matrices are

$$\mathbf{B}_1 = \begin{bmatrix} -0.5013 & 5.7726 & -5.1456 \\ -3.9060 & 2.1090 & 1.1683 \\ -6.1484 & 2.2873 & 4.4897 \end{bmatrix} \quad \mathbf{B}_2 = \begin{bmatrix} 0.0649 & 4.3379 & -4.2080 \\ -5.0660 & 2.5330 & 2.5330 \\ 0.3247 & -0.8269 & 1.4764 \end{bmatrix}$$

$$\mathbf{B}_3 = \begin{bmatrix} -0.5013 & -5.1456 & 5.7726 \\ 3.9060 & -1.1683 & -2.1090 \\ -6.1484 & 4.4897 & 2.2873 \end{bmatrix} \quad \mathbf{B}_4 = \begin{bmatrix} -2.2373 & 5.6955 & -2.9647 \\ -5.1645 & 4.4155 & 0.2555 \\ -6.0364 & -2.1052 & 8.6351 \end{bmatrix}$$

$$\mathbf{B}_5 = \begin{bmatrix} 0.2122 & 3.3592 & -2.9348 \\ -6.3662 & 3.1831 & 3.1831 \\ 0.2122 & -5.3011 & 5.7255 \end{bmatrix} \quad \mathbf{B}_6 = \begin{bmatrix} 2.5663 & 3.2937 & -5.3665 \\ -4.8355 & 0.5845 & 4.7445 \\ 6.3654 & -8.3061 & 2.4342 \end{bmatrix}$$

After deriving the six local state-space models of (5-20), we then define six fuzzy rules to smoothly combine them to form the overall fuzzy model. Six fuzzy rules are described as

Rule *i*: IF *x* is M_1^i and *y* is M_2^i, THEN $\dot{\boldsymbol{\eta}} = \mathbf{B}_i \dot{\boldsymbol{u}}$

where M_1^i, *i* = 1, 2, …, 6, is the fuzzy term of rule *i* corresponding to the premise variable *x*, $M_1^1 = M_1^2 = M_1^3$ = "around 0.015", $M_1^4 = M_1^5 = M_1^6$ = "around 0.075"; M_2^i, *i* = 1, 2, …, 6, is another fuzzy term of rule *i* corresponding to the premise variable *y*, $M_2^1 = M_2^4$ = "around –0.075" $M_2^2 = M_2^5$ = "around 0", $M_2^3 = M_2^6$ = "around 0.075". Since *z* is approximated at only one point, the transition between local models does not depend on *z*. Consequently, the premise variables are only *x* and y.

In order to enable the transitions among the six separate fuzzy rules, we propose the following membership functions. $\mu_{M_1^i}(x)$ and $\mu_{M_2^i}(y)$, *i* = 1, 2, …, 6, are grades of membership corresponding to the fuzzy terms M_1^i and M_2^i, respectively (please refer to Figure 5-6). They are defined as

$$\mu_{M_1^i}(x) = 1 - \frac{1}{1+\exp(-\frac{x-0.045}{0.0045})}, \quad i = 1,2,3 \tag{5-21}$$

$$\mu_{M_1^i}(x) = 1 - \mu_{M_1^1}(x), \quad i = 4,5,6 \tag{5-22}$$

$$\mu_{M_2^i}(y) = 1 - \frac{1}{1+\exp(-\frac{y+0.03}{0.0075})}, \quad i = 1,4 \tag{5-23}$$

$$\mu_{M_2^i}(y) = \frac{1}{1+\exp(-\frac{y-0.03}{0.0075})}, \quad i = 2,5 \tag{5-24}$$

$$\mu_{M_2^i}(y) = 1 - \mu_{M_2^1}(y) - \mu_{M_2^2}(y), \quad i = 3,6 \tag{5-25}$$

The membership functions for the local state-space models are then derived by

$$w_i(\boldsymbol{\eta}) = \mu_{M_1^i}(x)\mu_{M_2^i}(y), \quad i = 1,2,...,6 \tag{5-26}$$

where $w_i(\boldsymbol{\eta})$, i = 1, 2,…, 6, are the employed membership functions and they possess the following property $\sum_{i=1}^{6} w_i(\boldsymbol{\eta}) = 1, w_i(\boldsymbol{\eta}) \geq 0, \forall i$.

Here we consider the full state-feedback control instead of output-feedback control, thus, **C** = **I** which leads to $\boldsymbol{\Gamma} = \mathbf{I}$. So far, the fuzzy model is established by substituting the derived $\mathbf{A}_i$, $\mathbf{B}_i$, **C**, and $w_i(\boldsymbol{\eta})$ into (5-12) and (5-13) as

$$\dot{\boldsymbol{\eta}} = \sum_{i=1}^{6} w_i(\boldsymbol{\eta})\mathbf{B}_i\dot{\boldsymbol{u}} \tag{5-27}$$

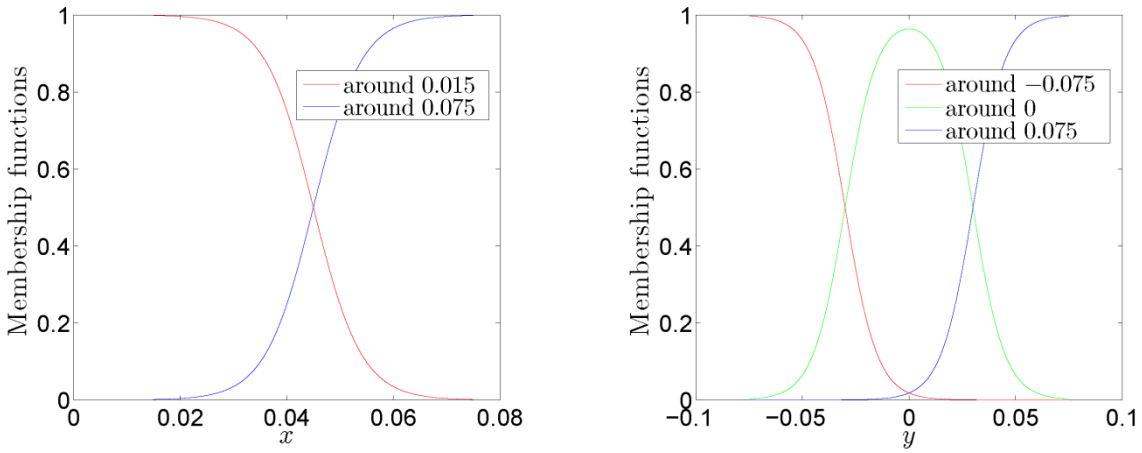


Figure 5-6 Illustrations of the employed membership functions. (Unit: horizontal axes: meter; vertical axes: ratio)

The difference of each entry between the original state-space model in (5-18) and the fuzzy model in (5-27) is measured by the mean absolute error (MAE) for all system states in the range of interest described in (5-19), which is defined by

$$\beta_{mn} = \frac{1}{N}\sum_{j=1}^{N}\left|\mathbf{J}_{mn}(\boldsymbol{\eta}_j) - \tilde{\mathbf{B}}_{mn}(\boldsymbol{\eta}_j)\right|, \quad m, n = 1,2,3, \tag{5-28}$$

where N is the number of a series of dense system states in the range of interest; m and n define the (m, n)-th entry of the corresponding matrix; $\boldsymbol{\eta}_j$ is the sampled system state; $\tilde{\mathbf{B}}(\boldsymbol{\eta}) = \sum_{i=1}^{6} w_i(\boldsymbol{\eta})\mathbf{B}_i$.

The calculated MAEs are β_{11} = 0.4149, β_{12} = 2.4850, β_{13} = 2.6373, β_{21} = 2.6893, β_{22} = 1.2943, β_{23} = 1.4764, β_{31} = 1.4137, β_{32} = 1.2130, β_{33} = 0.7695. It can be seen that the fuzzy modelling error exists due to the high nonlinearity of the original state-space model and the limited number of fuzzy rules, which can be further reduced in the future.

5.5.4 Fuzzy controller design

We first define two trajectory tracking cases to specify the reference model: one is to track a straight line in task space; another one is to follow an ellipse.

5.5.4.1 Straight line tracking

The straight line reference model in the form of (5-14) is given by

$$\dot{\boldsymbol{\eta}}_r = \mathbf{B}_r r \tag{5-29}$$

where $\mathbf{B}_r = [-0.0005, -0.002, 0.0018]^T$; $r = \begin{cases} 1, & t \le 60s \\ 0, & t > 60s \end{cases}$.

Given the above task, the corresponding fuzzy controller is designed by applying *Theorem* 5-1. Choosing the decision variables $\mathbf{X}$, $\mathbf{M}_j$, $\mathbf{N}_j$ as constant matrices; $\varepsilon_1 = \varepsilon_2 = 0.001$; $\sigma_1 = \sigma_2 = 0.1$ in (5-17); the feedback gains are obtained as follows

$$\mathbf{F}_1 = \begin{bmatrix} -1.1403 & -0.2225 & -0.3877 \\ -1.3084 & -0.3518 & -0.5492 \\ -1.0345 & -0.1597 & -0.6239 \end{bmatrix} \quad \mathbf{F}_2 = \begin{bmatrix} -0.4337 & 0.3401 & -0.0914 \\ -0.3560 & 0.2617 & -0.1716 \\ -0.3410 & 0.3552 & -0.2262 \end{bmatrix}$$

$$\mathbf{F}_3 = \begin{bmatrix} -2.7530 & -1.2628 & -1.8469 \\ -2.4707 & -1.6661 & -2.1170 \\ -2.4961 & -1.2549 & -2.4234 \end{bmatrix} \quad \mathbf{F}_4 = \begin{bmatrix} -0.4966 & 0.2297 & 0.5095 \\ -0.5153 & 0.1907 & 0.4838 \\ -0.4960 & 0.2428 & 0.4462 \end{bmatrix}$$

$$\mathbf{F}_5 = \begin{bmatrix} -0.2769 & 0.3449 & 0.0311 \\ -0.2992 & 0.2800 & -0.0013 \\ -0.2809 & 0.3165 & -0.0505 \end{bmatrix} \quad \mathbf{F}_6 = \begin{bmatrix} -0.9794 & 0.0313 & -0.8380 \\ 0.9314 & -0.1134 & -0.9170 \\ -0.9250 & 0.0159 & -1.0383 \end{bmatrix}$$

Here $\mathbf{G}_i \approx \mathbf{0}$, $i = 1, 2, \ldots, 6$, (the magnitude of the entries of the matrix $\mathbf{G}_i$ is less than 10^{-14}).

The designed fuzzy controller in the form of (5-16) can be acquired to comply with the fuzzy model in (5-27) as

$$\begin{aligned} \dot{\boldsymbol{u}} &= \sum_{j=1}^{6} w_j(\boldsymbol{\eta})\left(\mathbf{F}_j \Delta\boldsymbol{\eta} + \mathbf{G}_j \boldsymbol{\eta}_r\right) \\ \Delta\boldsymbol{\eta} &= \boldsymbol{\eta} - \boldsymbol{\eta}_r \end{aligned} \tag{5-30}$$

5.5.4.2 Ellipse tracking

The ellipse reference model in the form of (5-14) is given by

$$\dot{\boldsymbol{\eta}}_r = \mathbf{A}_r \boldsymbol{\eta}_r + \mathbf{B}_r r \tag{5-31}$$

where $\mathbf{A}_r = \begin{bmatrix} 0 & -0.1 & 0 \\ 0.1 & 0 & 0 \\ 0.2 & 0 & -0.1 \end{bmatrix}$; $\mathbf{B}_r = [0, -0.0045, -0.001]^T$; $r = 1$.

Given this task, the corresponding fuzzy controller is designed similarly by applying *Theorem* 5-1. Choosing the decision variables $\mathbf{X}$, $\mathbf{M}_j$, $\mathbf{N}_j$ as constant matrices; $\varepsilon_1 = \varepsilon_2 = 0.001$; $\sigma_1 = \sigma_2 = 1$ (different from the straight line tracking case) in (5-17); the feedback gains are then correspondingly obtained as follows:

$$\mathbf{F}_1 = \begin{bmatrix} -396.79 & -246.04 & -123.52 \\ -402.40 & -251.98 & -127.47 \\ -376.46 & -232.23 & -120.75 \end{bmatrix} \quad \mathbf{F}_2 = \begin{bmatrix} -69.131 & -35.498 & -22.311 \\ -65.949 & -35.047 & -22.464 \\ -51.417 & -24.213 & -18.871 \end{bmatrix}$$

$$\mathbf{F}_3 = \begin{bmatrix} -1226.8 & -774.00 & -395.00 \\ -1237.3 & -788.90 & -401.90 \\ -1200.5 & -757.50 & -395.80 \end{bmatrix} \quad \mathbf{F}_4 = \begin{bmatrix} -23.730 & -10.032 & 0.5280 \\ -36.800 & -18.550 & 3.9520 \\ -24.017 & -9.6064 & -0.4604 \end{bmatrix}$$

$$\mathbf{F}_5 = \begin{bmatrix} 5.5114 & 11.062 & 3.0908 \\ -12.779 & -1.1977 & -2.9270 \\ -8.3611 & 2.3123 & -2.2912 \end{bmatrix} \quad \mathbf{F}_6 = \begin{bmatrix} -361.56 & -219.25 & -119.59 \\ -378.63 & -232.62 & -125.81 \\ -366.06 & -222.23 & -123.85 \end{bmatrix}$$

$$\mathbf{G}_1 = \begin{bmatrix} 0.0648 & -0.0269 & -0.0436 \\ 0.0958 & -0.0428 & -0.0526 \\ 0.0967 & -0.0385 & -0.0598 \end{bmatrix} \quad \mathbf{G}_2 = \begin{bmatrix} 0.1497 & -0.0577 & -0.0982 \\ 0.1632 & -0.0655 & -0.1024 \\ 0.1661 & -0.0534 & -0.1064 \end{bmatrix}$$

$$\mathbf{G}_3 = \begin{bmatrix} -0.2025 & 0.0870 & 0.0874 \\ -0.2073 & 0.0869 & 0.1061 \\ -0.2210 & 0.0957 & 0.0961 \end{bmatrix} \quad \mathbf{G}_4 = \begin{bmatrix} 0.0543 & -0.0725 & -0.0403 \\ 0.0731 & -0.0787 & -0.0413 \\ 0.0885 & -0.0697 & -0.0516 \end{bmatrix}$$

$$\mathbf{G}_5 = \begin{bmatrix} 0.1556 & -0.0787 & -0.0683 \\ 0.1593 & -0.0859 & -0.0635 \\ 0.1789 & -0.0761 & -0.0766 \end{bmatrix} \quad \mathbf{G}_6 = \begin{bmatrix} 0.2876 & -0.1044 & -0.1492 \\ 0.2835 & -0.1058 & -0.1398 \\ 0.2807 & -0.1002 & -0.1455 \end{bmatrix}$$

The fuzzy controller in the form of (5-30) can be acquired but with feedback gains $\mathbf{F}_j$, $\mathbf{G}_j$, j = 1, 2,…, 6, designed for this ellipse trajectory.

5.6 Simulation examples and analysis of trajectory tracking tasks

We implement the proposed fuzzy controller in MATLAB simulation to investigate its performance. The simulation and control codes are attached in Appendix D. The simulation environment contains the aforementioned task space (please refer to (5-19)) with respect to a continuum manipulator. The manipulator's mathematical model described in (4-44) is utilised and `ode23` function command in MATLAB is executed to generate the continuous navigation path. In order to include the modelling inaccuracies and other real-time errors in simulation and validate the robust performance of the designed fuzzy controller, we introduce an additive term $\Delta\mathbf{J}$ to the analytically derived Jacobian matrix, i.e.

$$\dot{\boldsymbol{\eta}} = \left(\mathbf{J}(\boldsymbol{u}) + \Delta\mathbf{J}\right)\dot{\boldsymbol{u}} \tag{5-32}$$

Two different types of reference models respectively describing the straight line tracking trajectory and ellipse tracking trajectory are utilised in the simulation. To compare with other controllers, we implement all the controllers in the same situation, where the same additive term in the disturbed model is considered.

5.6.1 Straight line trajectory tracking task

In the simulation, the initial states of the disturbed model in (5-32) and the specified straight tracking trajectory in (5-29) are defined as $\boldsymbol{\eta}$ (0) = $\boldsymbol{\eta}_r$ (0) = [0.06, 0.06, 0.03]^T. The additive disturbance term is chosen as $\Delta\mathbf{J} = \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 \end{bmatrix}$ in the case. Implementing the designed fuzzy controller in (5-30), the simulation results are shown in Figure 5-7. We can see that the trajectory tracking task is effectively achieved by the proposed fuzzy controller.

We further compare our designed controller with three other types of controllers: (a) fuzzy controller with different H_∞ performance, (b) closed-loop Jacobian-based controller (Mahl, Hildebrandt, & Sawodny, 2014), (Siciliano, Sciavicco, Villani, & Oriolo, 2009) (please refer to Figure 5-8), and (c) open-loop Jacobian-based controller (Mahl, Mayer, Hildebrandt, & Sawodny, 2013) (please refer to Figure 5-9).

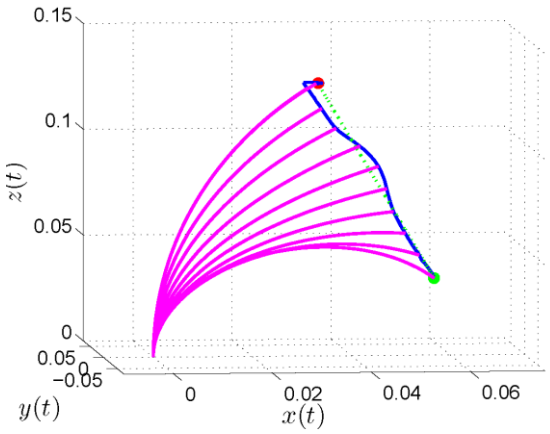


Figure 5-7 Performance of the designed fuzzy controller used to track a straight line trajectory (Unit: m). The trajectory of the continuum manipulator tip in 3D task space are captured and illustrated in blue line and the central backbone shape of the continuum manipulator is illustrated in pink; The green dot indicates the initial position [0.06, 0.06, 0.03]^T and the red dot indicates termination position [0.03, –0.06, 0.135]^T; the green dotted line shows the specified reference trajectory.

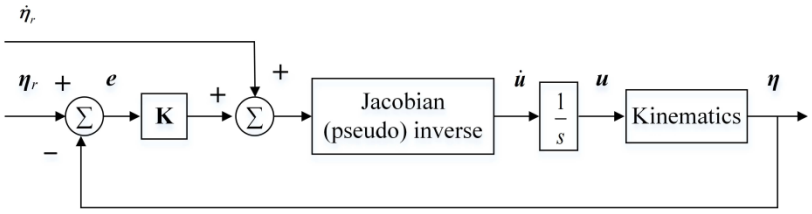


Figure 5-8 Overview of a closed-loop control that is based on the (pseudo-)inverse Jacobian method. Control input includes both the desired time-varying trajectory and the pre-planned task space velocity with respect to the desired task (command trajectory) (Mahl, Hildebrandt, & Sawodny, 2014), (Siciliano, Sciavicco, Villani, & Oriolo, 2009). **K** is a diagonal matrix and if **K** = **0**, then this control architecture becomes the same with the open-loop control illustrated in Figure 5-9.

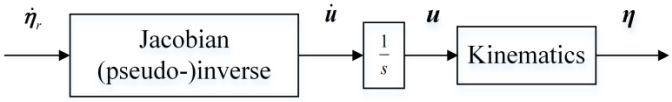


Figure 5-9 Overview of a feed-forward open-loop control architecture. Control input is the pre-planned task space velocity with respect to the desired task (command trajectory).

To design a fuzzy controller with different H_∞ performance, we choose $\sigma_1 = \sigma_2 = 100$ in (5-17) while other controller design parameters remain the same in *Theorem* 5-1. Therefore, the corresponding feedback gains can be similarly obtained; results are shown below:

$$\mathbf{F}_1 = \begin{bmatrix} -0.7876 & -0.1734 & -0.3090 \\ -0.7319 & -0.2653 & -0.4157 \\ -0.7302 & -0.1425 & -0.4870 \end{bmatrix} \quad \mathbf{F}_2 = \begin{bmatrix} -0.4261 & 0.1878 & -0.1786 \\ -0.3562 & 0.1164 & -0.2449 \\ -0.3460 & 0.2016 & -0.2930 \end{bmatrix}$$

$$\mathbf{F}_3 = \begin{bmatrix} -1.7733 & -0.7787 & -1.0933 \\ -1.5753 & -1.0420 & -1.2604 \\ -1.5934 & -0.7661 & -1.4639 \end{bmatrix} \quad \mathbf{F}_4 = \begin{bmatrix} -0.4690 & 0.1127 & 0.2795 \\ -0.4807 & 0.0604 & 0.2790 \\ -0.4687 & 0.1007 & 0.2311 \end{bmatrix}$$

$$\mathbf{F}_5 = \begin{bmatrix} -0.2694 & 0.1912 & -0.0536 \\ -0.2759 & 0.1313 & -0.0745 \\ -0.2536 & 0.1635 & -0.1175 \end{bmatrix} \quad \mathbf{F}_6 = \begin{bmatrix} -0.7860 & -0.0819 & -0.7938 \\ -0.7555 & -0.1895 & -0.8512 \\ -0.7385 & -0.0871 & -0.9315 \end{bmatrix}$$

Here $\mathbf{G}_i \approx \mathbf{0}$, $i = 1, 2, \ldots, 6$, (the magnitude of the entries of the matrix $\mathbf{G}_i$ is less than 10^{-8}).

The closed-loop Jacobian-based controller is designed based on Equation (40) in (Mahl, Hildebrandt, & Sawodny, 2014) with **W** = **I**, **K** = 0.1 **I** resulting $\dot{\boldsymbol{u}} = \mathbf{J}(\boldsymbol{u})^{-1}\left(\dot{\boldsymbol{\eta}}_r + 0.1(\boldsymbol{\eta}_r - \boldsymbol{\eta})\right)$. The open-loop Jacobian-

based controller is simply given by $\dot{\boldsymbol{u}} = \mathbf{J}(\boldsymbol{u})^{-1}\dot{\boldsymbol{\eta}}_r$. The comparisons among the total four controllers are illustrated in Figure 5-10 and Figure 5-11. The proposed fuzzy controller demonstrates the best performance with the minimum tracking errors. Compared with the additional fuzzy controller with different H_∞ performance, the results imply that the smaller the values of σ_1 and σ_2, the better the H_∞ performance tracking performance governed by (5-17). The open-loop and closed-loop suffer from the modelling inaccuracies. Although the closed-loop controller can reduce the tracking error based on the real-time feedback information, large modelling error results in poor performances. Both Jacobian-based controllers need online updates of the Jacobian which causes a computational burden, which could be particularly problematic in a real-time system; on the other hand, our fuzzy controllers are very efficient and have a low computational load.

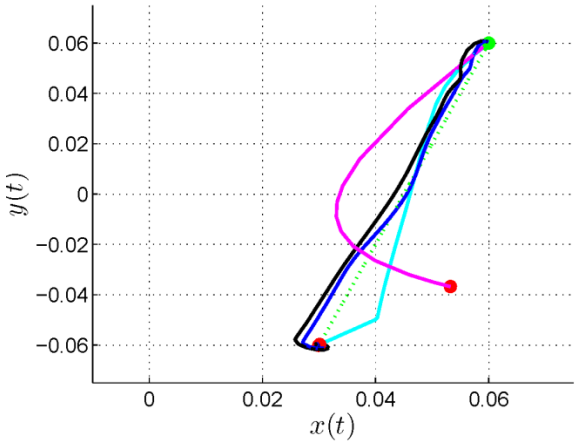


Figure 5-10 Performance comparisons of the four controllers shown via the *xy* plane viewof 3D task space. The green dotted line shows the specified reference trajectory (Unit: m). The blue, black, cyan, and pink trajectories indicate the trajectories based on the proposed fuzzy controller, the fuzzy controller with different H_∞ performance, the closed-loop Jacobian-based controller, and the open-loop Jacobian-based controller, respectively.

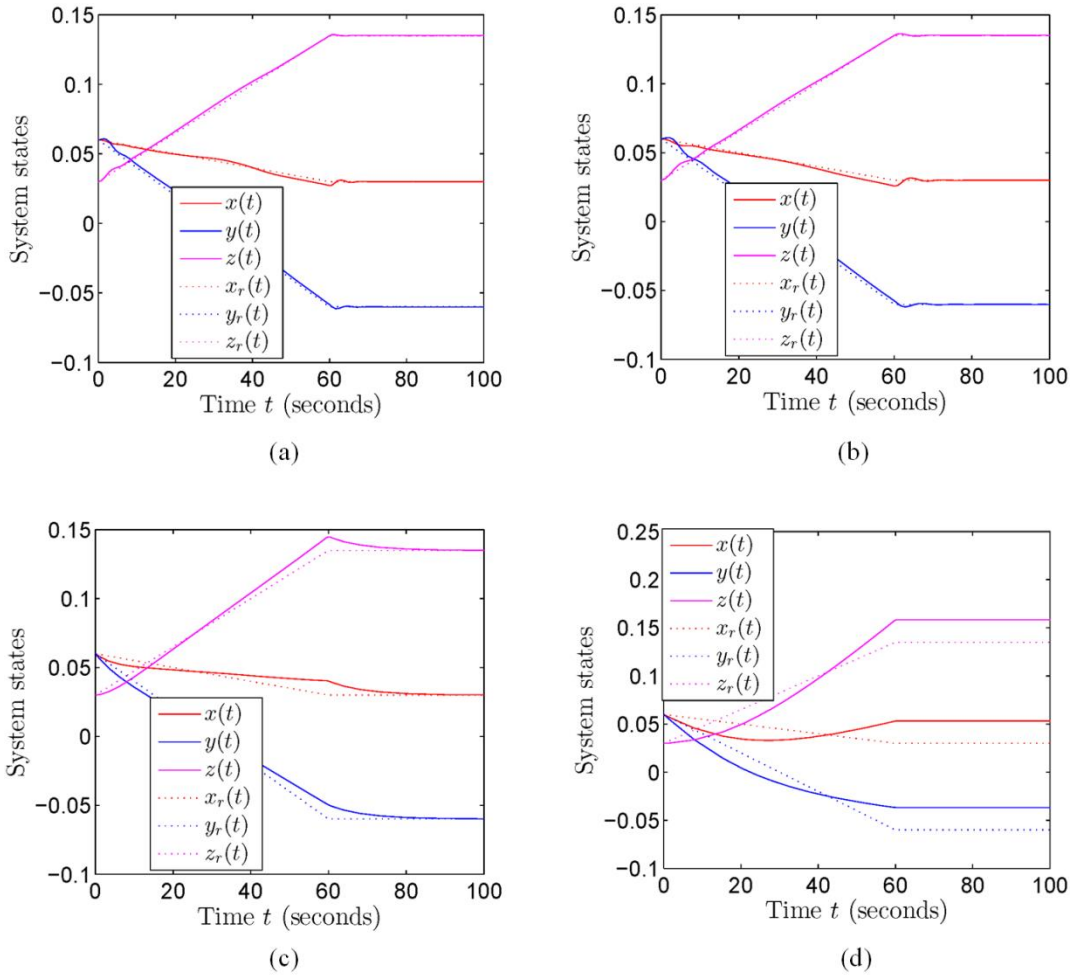


Figure 5-11 Illustrations of time responses with respect to each of the four controllers: (a) the proposed fuzzy controller, (b) the fuzzy controller with different H_∞ performance, (c) the closed-loop Jacobian-based controller, and (d) the open-loop Jacobian-based controller, respectively.

The numerical comparison index is calculated by using the integral absolute error (IAE) $\hbar = \int_0^{t_f} \left(|x - x_r| + |y - y_r| + |z - z_r| \right) dt$, where t_f = 100 seconds. The results are shown as in Table 5-1 and it further illustrates the superiority of the proposed fuzzy controller.

Table 5-1 Numerical comparison of different controllers' straight line tracking performances via IAE.

Controllers	***IAE***
Proposed fuzzy controller	0.3513
Fuzzy controller with different H_∞ performance	0.4279
Closed-loop Jacobian-based	0.9997
Open-loop Jacobian-based	4.6803

5.6.2 Ellipse trajectory tracking task

In this simulation, the initial states of the disturbed model in (5-32) and the ellipse tracking trajectory in (5-31) are $\boldsymbol{\eta}(0) = \boldsymbol{\eta}_r(0) = [0.065, 0, 0.1]^T$. The additive disturbance term, 3×3 matrix $\Delta\mathbf{J}(t)$, is chosen by assigning each of all 9 entries as $0.1\sin(\frac{\pi t}{30})$. Implementing the designed fuzzy controller in (5-30) with the feedback gains derived in 5.5.4 for the ellipse trajectory case, the simulation results are shown in Figure 5-12. The ellipse tracking task is accomplished perfectly by the designed fuzzy controller.

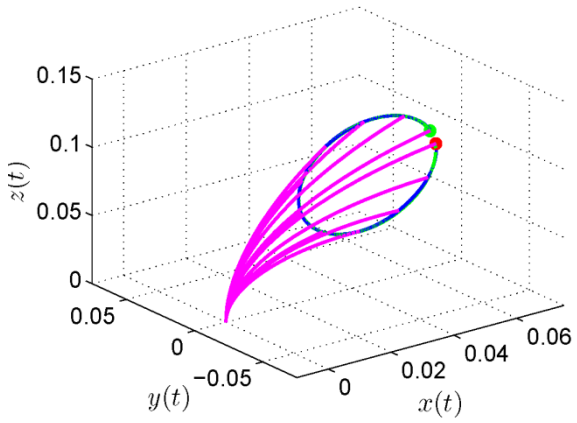


Figure 5-12 Performance of the designed fuzzy controller used to track an ellipse trajectory (Unit: m). The trajectory of the continuum manipulator tip in 3D task space are captured and illustrated in blue line starting from the initial position $[0.065, 0, 0.1]^T$ coloured in green to termination position coloured in red; The central backbone shape of the continuum manipulator is illustrated in pink; the green dotted line shows the specified reference trajectory.

Here we also compare our designed fuzzy controller with three other types of controllers: closed-loop Jacobian-based controller and open-loop Jacobian-based controller are same as those used for straight line trajectory tracking task; another linear controller is designed with the same methodology as to design the fuzzy controller but choosing only one operating point $\boldsymbol{\eta} = [0.04, 0, 0.075]^T$ (a special case of the fuzzy model). The linear controller is described as

$$\dot{\boldsymbol{u}} = \mathbf{F}\Delta\boldsymbol{\eta} + \mathbf{G}\boldsymbol{\eta}_r \tag{5-33}$$

where $\mathbf{F} = \begin{bmatrix} -0.8497 & 0.0150 & 0.2688 \\ -0.8516 & -0.0198 & 0.2825 \\ -0.8484 & -0.0197 & 0.2554 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} -0.0762 & -0.1153 & 0.0363 \\ -0.0750 & -0.1155 & 0.0384 \\ -0.0669 & -0.1152 & 0.0343 \end{bmatrix}$.

The comparisons of the four controllers are illustrated in Figure 5-13 and Figure 5-14. Similar results with those for straight line tracking case can be obtained. We can see that although the actual trajectories from both the open-loop and closed-loop Jacobian-based controllers are ellipse-like, they quickly run away from the defined the ellipse after the starting position. The performance of the linear controller obtained from one operating point is worse than the fuzzy controller obtained from six operating points. One operating point is not enough to represent the original nonlinear model. The numerical comparison results using IAE index is given in Table 5-2.

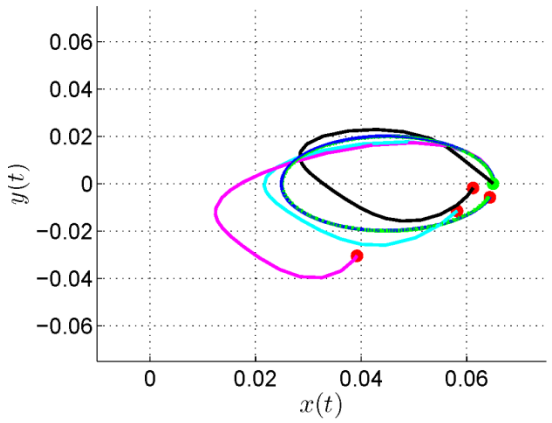


Figure 5-13 Performance comparisons of the four controllers with the *xy* plane view of 3D task space (Unit: m). The green dot indicates the initial position and the red dot indicates termination position; the green dotted line shows the specified reference trajectory. The blue, black, cyan, and pink trajectories indicate the trajectories based on the proposed fuzzy controller, the linear controller with one single operating point, the closed-loop Jacobian-based controller, and the open-loop Jacobian-based controller, respectively.

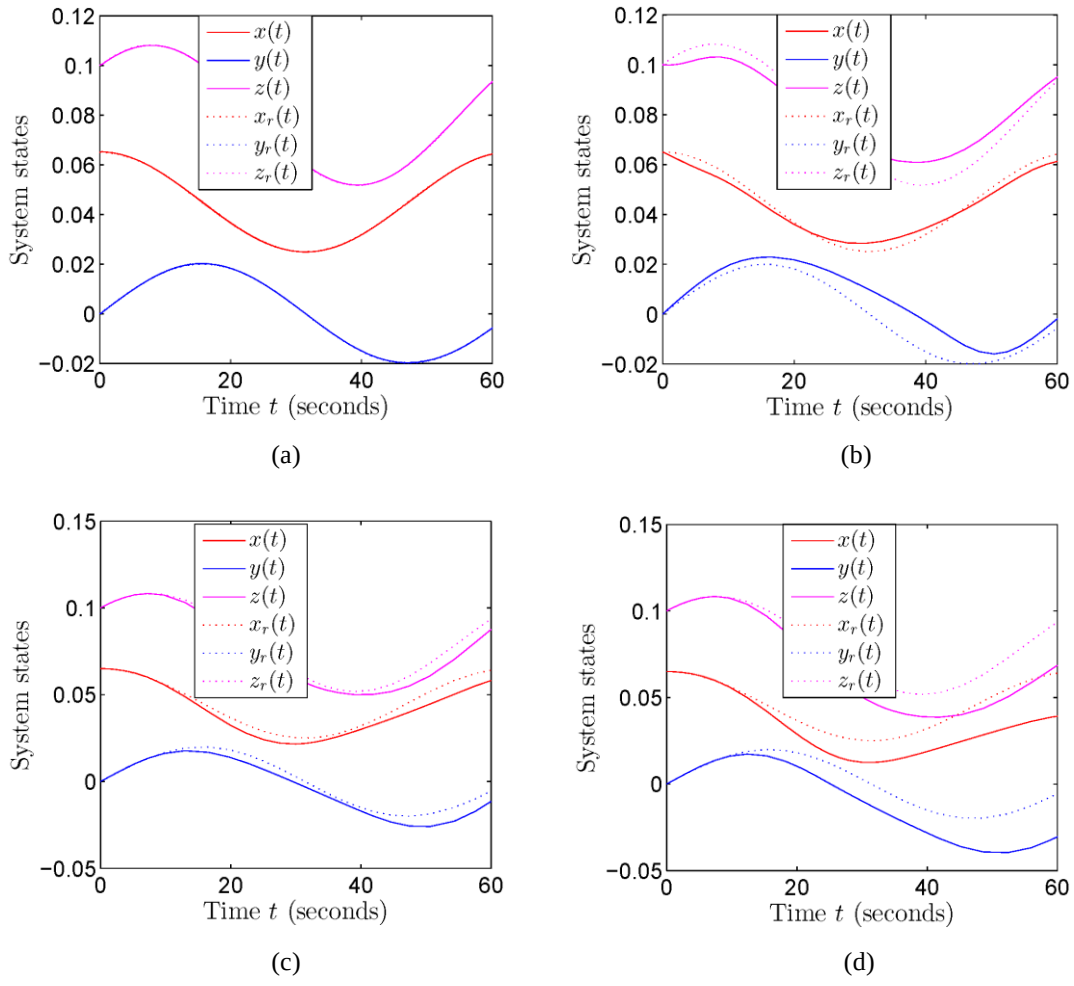


Figure 5-14 Illustrations of time responses with respect to each of four controllers: (a) the proposed fuzzy controller, (b) the linear controller with one single operating point, (c) the closed-loop Jacobian-based controller, and (d) the open-loop Jacobian-based controller, respectively.

Table 5-2 Numerical comparison of different controllers' ellipse trajectory tracking performances via IAE.

Controllers	***IAE***
Proposed fuzzy controller	0.0214
Linear controller	0.7800
Closed-loop Jacobian-based	0.6271
Open-loop Jacobian-based	2.0268

5.7 Experiments of multi-tendon driven continuum manipulator tracking tasks

The proposed fuzzy controller is implemented on a tendon-driven continuum manipulator, whose design was presented in Chapter 3. This continuum manipulator demonstrates an effectively decoupled bending with contraction via three tendons at the periphery, thus, in line with the previously presented kinematic model for a general continuum manipulator. The contraction capability enables the length of the manipulator to vary from the original full length to contract to a length of about 70%. In order to measure the manipulator's tip position, a commercial electromagnetic (EM) tracking system NDI Aurora® is used. One 0.8mm diameter × 11mm length sensor coil is integrated in the head of the continuum manipulator to track real-time tip positions and orientations. Each tendon is actuated via a DC motor (Maxon Motor®). For comparative purposes, both the traditional Jacobian-based open-loop and closed-loop controllers are also implemented under the same condition; Figure 5-15 illustrates the experimental setup.

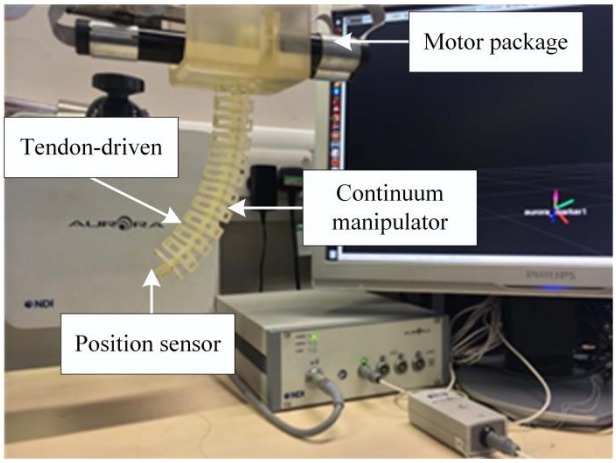


Figure 5-15 Experimental setup. The controllers are implemented on a tendon-driven continuum manipulator with validated constant-curvature bending performance. The NDI Aurora® EM position sensor is used for the purpose of tracking the manipulator tip.

5.7.1 System description

All the three controllers as well as the reference trajectory generator are implemented in the Robot Operating System (ROS) Environment on an Intel® Core i3 @ 2.40 GHz and 1.5GB RAM based platform running Linux Mint 13 Operating System. The control signal is fed to the EPOS2 Module motor controller via USB port to control the velocities of three Maxon DC Motors, each equipped with encoders

to ensure precise velocity control. The motors are connected to the tendons via a gearbox with a reduction ratio of 128:1. The rotation of the motors moves each tendon with the desired velocity. A change in tendon length will move the tip of the continuum manipulator; the motor control is based on the kinematic model.

An Aurora sensor coil, embedded in the tip of the manipulator, will give the position of the tip with respect to another sensor coil embedded in the base of the manipulator. This information is fed to the computer via the NDI Aurora tracking system and used as a feedback to the fuzzy controller. A standard Jacobian-based closed-loop controller receives Aurora signals in the same way during the comparative experiments. The reference trajectory in this experiment is chosen to be a straight line measured with respect to the base of the robot. The reference trajectory position, as well as the manipulator tip's position and the tendon velocity as control signal are also recorded via ROS to enable further analysis and documentation. The block diagram of the experimental system integration is shown in Figure 5-16. The interconnections of the ROS setup for experiments are illustrated using a ROS graph, Appendix E. The parameters of the experiment is shown Table 5-3.

The control programs are first tested off-line using simulated tracking sensor feedback. These tests validate the advantage of the fuzzy controller in terms of its reduced amount of necessary calculations. When executed in our computer system, the proposed fuzzy controller operates at an execution rate of 168.79Hz (0.0059 seconds execution time per iteration) as opposed to 40.90Hz (0.0244 seconds execution time per iteration) for the Jacobian-based methods. This big gap in execution time is caused by the fact that the fuzzy controller's feedback gains do not need to be updated. However, the Jacobian-based controllers need to do numerical integrations to estimate the current length of each tendon and update the Jacobian matrix in every step of the control executions. Besides, the matrix inversion operation – a complex mathematical process – to inverse the derived Jacobians is needed in the Jacobian-based controllers and slows down the computations. Based on the analysed execution rate above, the system is thus determined to be executed with a sampling rate of 40Hz (which is ensured to be less than the rate of 40.90Hz).

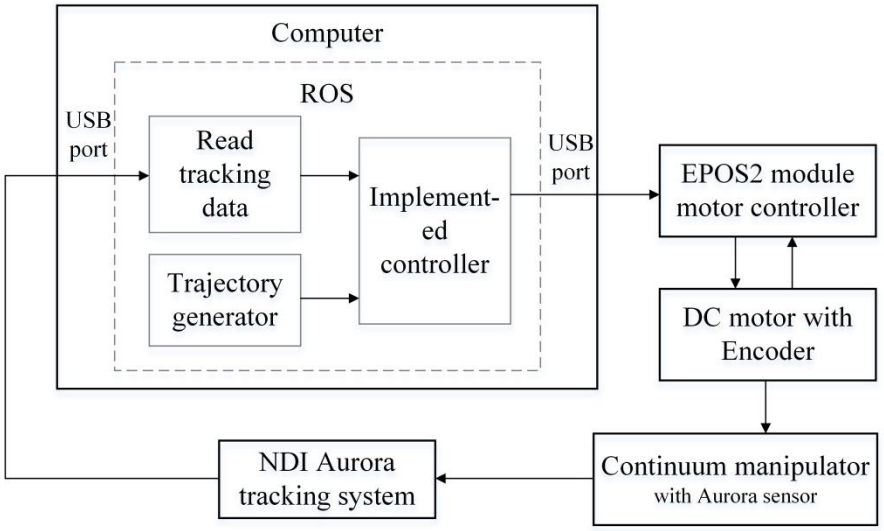


Figure 5-16 Block diagram of the experimental system integration.

Table 5-3 Experimental parameter data settings (unit: mm).

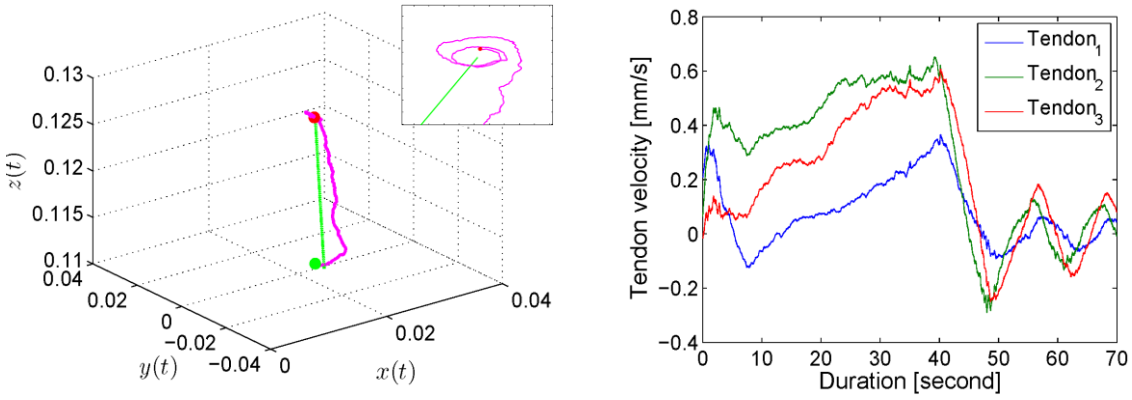
Description	***value/range***
Full length of the continuum manipulator	143 mm
Cross-section radius of the continuum manipulator (d)	13.4 mm
Specified workspace range*	x: [7, 31.5] y: [110, 134] z: [110, 134]
Velocity ($\mathbf{B}_r$)	$[0.250, 0.750, 0.267]^T$
Duration	70 seconds (after 40 seconds the reference trajectory terminates.)

*The global coordinate system is established on the bottom of the continuum manipulator as shown in Figure 4-14(b). The reference model is a straight line starting from the specified initial position to the termination position as given above.

5.7.2 Experimental results and analysis

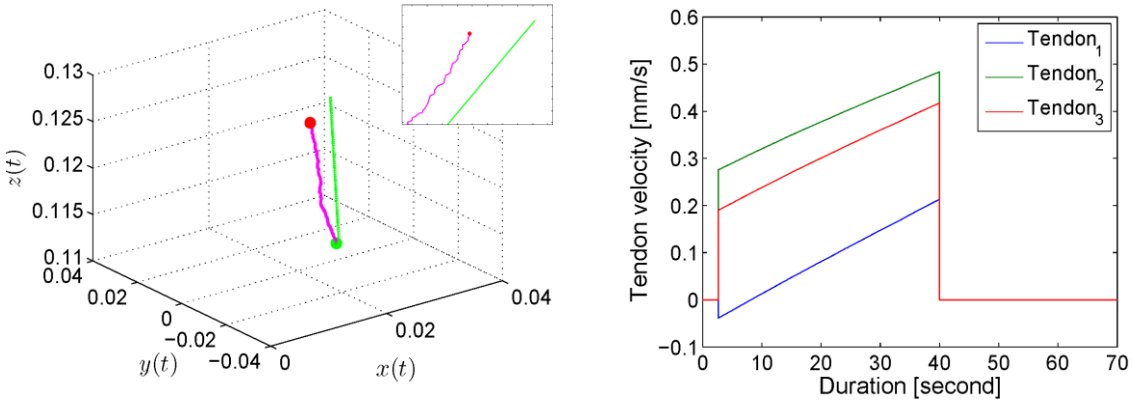
To implement the fuzzy-model-based control for experimental studies, six different local operating points [7, –31.5, 122], [7, 0, 122], [7, 31.5, 122], [31.5, –31.5, 122], [31.5, 0, 122], [31.5, 31.5, 122], (Unit: mm) with respect to the specified workspace range in Table 5-3, are chosen to approximate the state-space model. Accordingly, membership functions are then derived and utilised for both fuzzy model

construction and fuzzy controller design. The experimental results are illustrated in Figure 5-17. Despite the fact that model discrepancies and hardware tolerances exist, the proposed fuzzy-model-based approach still accomplishes the tracking task. As shown in Figure 5-17(a) the final stage of the experimental recorded trajectory presents a converged spiral, which indicates the feasibility of the controller. For comparison purposes, we also implemented two other traditional Jacobian-based controllers and tested in an experimental study. Figure 5-18 shows tracking performance with an open-loop Jacobian-based controller, whose control architecture is shown in Figure 5-9; Figure 5-19 shows the tracking performance with a closed-loop Jacobian-based controller ($\mathbf{K} = 0.1\ \mathbf{I}$), whose control architecture is shown in Figure 5-8. From Figure 5-18 and Figure 5-19, we can see that both traditional controllers achieve the tracking tasks but there exist a significant distance between the end-point to the target. Based on Figure 5-18(a) and Figure 5-19(a), the performance with the closed-loop controller is better than the performance with the open-loop controller. The open-loop control execution leads to an accumulation of the tracking errors, and it can be seen that the experimental recorded trajectory gradually moves away from the reference without any trend to decrease the error. The closed-loop Jacobian-based controller keeps to a trajectory that is almost parallel with respect to the reference. After 40 seconds when the reference model terminates at the target point, the closed-loop control will drive the tip of the manipulator to gradually approach the steady target, while the open-loop control terminates at exactly 40 seconds. Due to the delay on the ROS Node initialization, the controller does not start to produce control signals immediately, rather it lags by a small duration of time at the beginning of the experiments. The numerical comparisons regarding the performances with these three different controllers are given in Table 5-4. It can be seen that, with regards to the targeting precision and the integral absolute error (IAE) the proposed fuzzy-model-based controller shows advantages. It is also to be noted that the performance of implementing the open-loop controller reflects the accuracy of our kinematic model and the hardware shortcomings. Both the closed-loop Jacobian based controller and the proposed fuzzy-model-based controller still have space to be further improved so that a better tracking performance can be expected. These experiments in this paper validate the feasibility of the fuzzy-model-based controller to be implemented for continuum manipulators with some appealing advantages. This is the first work of achieving task space closed-loop control with a fuzzy-model-based approach.



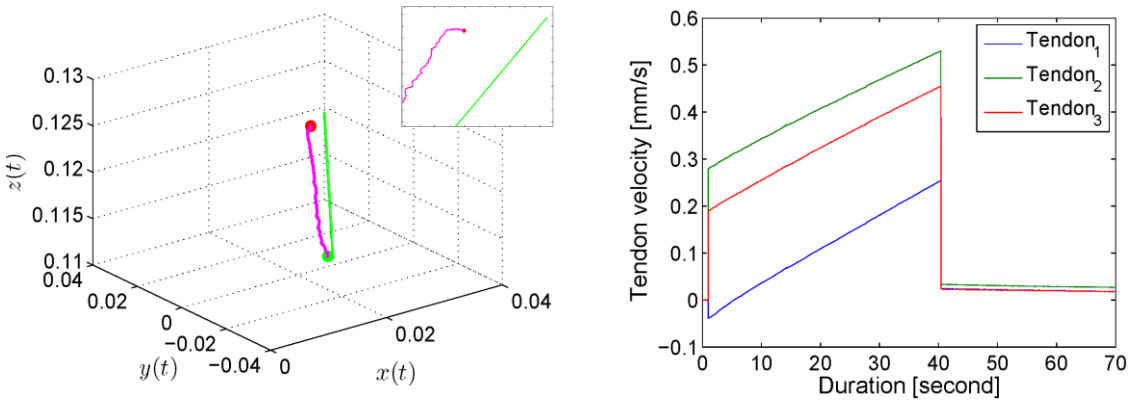
(a) (b)

Figure 5-17 The experiment of a trajectory tracking execution via fuzzy-model-based approach. (a) shows the experimental recorded trajectory in 3D space (Unit: m) and the zoom-in view around the target. The green dot indicates the initial position and red dots indicate the termination positions. The green line shows the specified reference trajectory. The magenta trajectory indicates the trajectory based on the proposed fuzzy controller. (b) shows the tendon speed control signals.



(a) (b)

Figure 5-18 The experiment of a trajectory tracking execution via open-loop Jacobian-based approach, corresponding to the controller illustrated in Figure 5-9. (a) shows the experimental recorded trajectory in 3D space (Unit: m) and the zoom-in view around the target. The green dot indicates the initial position and red dots indicate the termination positions. The green line shows the specified reference trajectory. The magenta trajectory indicates the trajectory based on the open-loop Jacobian-based controller. (b) shows the tendon speed control signals.



(a) (b)

Figure 5-19 The experiment of a trajectory tracking execution via closed-loop Jacobian-based approach, corresponding to the controller illustrated in Figure 5-8. (a) shows the experimental recorded trajectory in 3D space (Unit: m) and the zoom-in view around the target. The green dot indicates the initial position and red dots indicate the termination positions. The green line shows the specified reference trajectory. The magenta trajectory indicates the trajectory based on the closed-loop Jacobian-based controller. (b) shows the tendon speed control signals.

Table 5-4 Numerical comparison of different controllers tacking performances in experiments.

Controllers	***Targeting precision***	***IAE***
Proposed fuzzy controller	0.7165 mm	9.7831
Closed-loop Jacobian-based	3.1 mm	11.8707
Open-loop Jacobian-based	5.2 mm	17.2440

5.8 Discussions on trajectory tracking control

A fuzzy controller has been proposed for autonomous execution of end-effector trajectory tracking tasks of a continuum manipulator, overcoming model complexities and uncertainty issues that plague other types of controllers. We utilise the very general constant-curvature approximation technique for kinematics derivation and achieve a very satisfactory control synthesis using a fuzzy-model-based approach. Two specified reference models are provided to test the tracking performance of the proposed controller. In MATLAB simulations, the proposed controller was implemented and compared with three other controllers. In order to include the modelling inaccuracies and other real-time errors in simulation and validate the performance robustness of the designed fuzzy controller, we artificially introduced noise

disturbing the simulated sensor signals, representing the real situation as closely as possible. The results show that the designed fuzzy controller has the best performance with regards to the minimum tracking errors and effectively accomplishes both tracking tasks. The other Jacobian-based controllers suffer from model inaccuracies.

In addition, experiments were conducted employing a rapid-prototyped continuum manipulator whose model is in line with the one derived. The assigned test task was to track a straight line in 3D task space and the experimental results verified the feasibility of the controller in presence of modelling discrepancies and hardware tolerances. Several advantages in terms of computational cost, targeting precision and overall performance compared to Jacobian-based controllers are presented based on experiments.

Future work will include further refining the controller design and testing the controller with practical continuum manipulator systems. Multi-section continuum manipulators with more degrees of freedom are to be analysed and controlled with the fuzzy-model-based approach. We also seek to utilise the fuzzy-model-based methodology to achieve the “whole arm manipulation” via incorporating more sensing information and dynamic model of the manipulator, and simultaneously controlling multiple virtual-joints along the flexible backbone.

Chapter 6 Conclusion and Future Work

This thesis has proposed a novel design of continuum manipulators in the actively expanding field of continuum robots. Robots created under this paradigm (using multilayer planar springs) possess several advantages, which were presented and experimentally validated in the previous chapters. Also, the modelling and control strategies with respect to the proposed manipulator have been investigated. Under the constant-curvature assumption, the kinematic models for the continuum manipulator with the varying length and 3D bending abilities have been established. Finally, motion control of the continuum manipulator was studied and fuzzy control strategies were implemented for both set-point regulation and trajectory tracking tasks. The work presented in this thesis laid the foundations in continuum manipulator design, modelling, intelligent control for continuum robots, and on this basis, there are several promising research topics, which could further expand the boundaries of the field.

6.1 Conclusion

This thesis is centred on the novel design of the continuum manipulator using serially connected multilayer planar springs. This continuum manipulator is inherent with the linear and decoupled bending and contraction. These features are the crucial and highly desired properties for the physical structure of continuum manipulators. After proposing the design of the continuum manipulator, mathematical modelling is particularly important to coordinate its movements. A unique and efficient compliance analysis method was presented. Subsequently, the derived compliance matrix revealed several important characteristics underlying the planar spring and the physical structure of the continuum manipulator. They were further validated with both FEA simulations and experiments. The advancement in aspect of robotic control of continuum manipulators is relatively limited in the early works. In this thesis, the fuzzy control strategies were implemented on the motion control of continuum manipulators. Superior performances compared to classical Jacobian-based controllers were demonstrated. The detailed summaries of results are presented as follows.

6.1.1 Conceptual design of continuum manipulators

The design of our proposed continuum manipulator was introduced in Chapter 3. Firstly the related works of the continuum manipulator design were reviewed. We started from looking at the hyper-redundant manipulator and recalled the conceptual origin of continuum-style robots. Different designs from this origin are categorised according to their distinct backbone structures. Continuum robots come as a branch parallel with hyper-redundant robots. It has to say that many hyper-redundant robots really look like continuum robots with continuous external appearances. However, we point out that continuum robots are those having the continuously bending and infinite-degrees-of-freedom features embodied in elastic structures with inherent compliance. Thus, continuum robots are all under-actuated. Frequently applied designs of continuum robots are spring-based backbone design, elastic-rod/tube-based backbone design, and soft continuum robot design.

Our design of the continuum manipulator utilises the local spring concept. The rotational and translational motions of this type of planar springs are for the first time combined together for investigations and applied to generate the cumulative bending and contraction for continuum robot. In the past, this type of planar springs was found in applications almost only related to their translational motions of the central platform. Furthermore, we also designed the double-layer module consisting of two opposite positioned planar springs in our prototype for 3D printing. The consideration of this design is to

simplify the 3D prototyping and to allow the length of the continuum manipulator to cope with various intended scenarios. The current prototype is articulated by three tendons in the periphery. Alternatively, four tendons can also achieve the functions of bending and contraction, but need one extra motor. This prototype can be considered as one-section design as all the three tendons are terminated in one level which only results in one curve bending.

The planar spring employed in the prototype is the standard radial-leg design. Also, design variations of such U-shape planar springs were investigated. They are summarised as three types: radial-leg design, side-leg design, and titled-leg design. With increasing the angle of leg arrangement, the diameter of the outer base increases. More importantly, their corresponding compliance properties will be changed by varying the angle of leg arrangement, and this was analytically studied after deriving the compliance matrix in Chapter 4.

Experiments were first conducted on the single module and three-module assembly with a linear guide and a force sensor. Both bending motions with one-tendon actuation and contraction motions with three-tendon actuation were performed. The tension of the taut tendon was recorded. The experimental results revealed the linear bending and contraction motions. Besides, the bending motion is effectively decupled with contraction (the maximum coupled contraction in experiments was as small as 4% of the original length, thus it could be considered that the length was invariant). The tested planar spring modules are fabricated by a 3D-printer with ABS materials. Due to the properties of plastic materials, the tested modules exhibit hysteresis behaviours, and it is expected that using other more low-hysteresis materials, for example super-elastic NiTi, could reduce the hysteresis in practise.

The full continuum manipulator was also tested. The contraction ratio of the currently fabricated prototype is 0.7. And as expected the robot effectively decreases the contraction as well when generating bending deflections. In experimental tests, only 5.6% of the original length was contracted when the continuum manipulator bended until an angle of 160˚. Due to the varying length of this continuum manipulator, the enlarged workspace was demonstrated. Compared to the helical-spring-based continuum manipulator, the proposed continuum manipulator exhibits both decoupled bending and contraction and broadening working areas.

6.1.2 Mathematical modelling and analysis

In Chapter 2, the classical mathematical tools to study the mechanics of conventional rigid-link robots were presented, laying foundations in mathematical modelling for continuum manipulators. We started

from the coordinate transformations which were presented based on Lie group, Lie algebra and screw theory. The concepts of screws, twists, and wrenches were presented and they were introduced in the form of one element of a six-dimensional vector space. They would be utilised to derive the compliance matrix of the proposed continuum manipulator. Particularly, the transformations of twists and wrenches between coordinate frames – adjoint transformations – were explicitly presented. In the second part of Chapter 2, the fundamental knowledge of differential geometry of cures was presented. Frenet-Serret (F-S) coordinate frame was introduced and under this framework the relative position and orientation information of a local F-S frame attached to an arbitrary point along the curve was derived.

In Chapter 4, the mechanics of continuum manipulators was investigated. Firstly six-dimensional compliance characteristics of the planar spring were investigated using a compliance matrix based approach. A twist deflection of the Lie algebra $se(3)$ of Lie group $SE(3)$ and a wrench as an element of the dual Lie algebra $se^*(3)$ were introduced based on the knowledge presented in Chapter 2. The planar spring was treated as a type of hybrid flexure mechanism. Based on the fundamental compliance matrix of a rectangle beam that was derived by other researchers, the compliance matrix of the planar spring was then developed with a bottom-up approach. This developed compliance matrix is diagonal, which further reveals the decoupling and isotropism characteristics of the planar spring. We also numerically compared the group translational elements and the group of rotational elements, which shown that three major elements were much larger than other three elements in the compliance matrix. Therefore, the following discussions were only focused on these three major elements respectively corresponding to one translation DOF and two rotational DOFs.

By integrating the varying angle of leg arrangement into the compliance matrix derivation, a general compliance matrix of design derivations of the planar spring was derived. This matrix led to the study of the changing compliance of planar springs due to the change of the angle of leg arrangement. Therefore, we not only knew that swinging the legs of the planar spring would change its outer diameter, but also would change its compliance properties. We also numerically compared the compliance of different configurations of planar springs. Regarding the translational compliance elements, their values in all the three different configurations (the angle of leg arrangement is 0˚, 90˚, 150˚, respectively) are the same. The change of two major rotational compliance elements, with respect to the angle of leg arrangement from 0˚ to 90˚, was significant. The value of rotational compliance element of the 0˚ configuration is 61.41rad/(N·m), and comparably that of the 90˚ configuration is 17.8rad/(N·m) (for the 90˚ configuration, that is 15.61rad/(N·m)). In all configurations from 0˚ to 90˚, the major compliance elements are much

larger than the one left minor compliance (in the 0˚ configuration, it is 8.4 times). The analytical compliance analysis based on the derived compliance was then validated via FEA simulations.

The behaviour of the full continuum robot was discussed, including its compliance, kinematics, and statics. Based on the classical constant-curvature assumption, the kinematics of the continuum manipulator was developed. According to different practical applications, two different kinematic models corresponding to three-tendon-driven and single-tendon-driven continuum manipulators were presented, which were utilised in Chapter 5 to develop control strategies for different tasks.

To complete the mathematical modelling of the continuum manipulator, the dynamic models were developed. Here the kinematics was established under the Frenet-Serret frame and the Euler-Lagrangian equation of motion was applied. The dynamics for our proposed continuum manipulator includes the analysis of all the kinetic energy, gravitational energy, and elastic energies due to both bending and contraction.

6.1.3 Fuzzy control strategies for continuum manipulators

In Chapter 5, two kinds of fuzzy control strategies were implemented for motion control of continuum manipulators. Both cases belong to the task space closed-loop control and the real-time tip position information is assumed to be provided by an attached sensor.

The first control strategy is the fuzzy logic controller (or called fuzzy-model-free controller in contrast to the second implemented fuzzy-model-based controller). This controller was designed for set-point regulation tasks. Recalling that in Chapter 4 we have derived kinematic models for each of three-tendon-driven and single-tendon-driven (with a rotatory platform for 3D-space positioning) continuum manipulators, this control case was focused on the latter kinematic model, and the corresponding continuum manipulators have appeared for some medical applications. This kinematic model was first converted to the state-space model form for controller design and then linearized at six operating points within the manipulator's workspace. Six linear state-feedback controllers were designed for six local models after the linearization. When designing these controllers, two different methods were used. They are linear-quadratic regulator (LQR) method and pole placement method. Then a fuzzy logic controller having six rules was proposed to combine the developed six linear controllers. The performance was simulated via MATLAB and a comparison between the LQR-based optimised controller and the non-optimised controller was presented. Both controllers completed the set-point regulation task.

The second control strategy is the fuzzy-model-based controller. This controller was designed for trajectory tracking tasks. This control method is based on stability analysis for general continuous-time nonlinear systems and membership functions were also utilised to the linearized state-space models for blending. The kinematic model in this case corresponded to the three-tendon-driven continuum manipulators (for example, our proposed continuum manipulator) and the same local approximation technique as in the previous case was utilised as well. The performance of the fuzzy-model-based controller was first investigated in MATLAB simulations by following both the straight line trajectory and the ellipse trajectory. Furthermore, we compared this fuzzy-model-based controller with other classical Jacobian-based controllers that have been found in literature for motion control of continuum manipulators. In MATLAB simulations, the modelling inaccuracies and real-time errors were included to simulate the real situation as closely as possible and the fuzzy-model-based controller exhibited the best performance. It has the minimum tracking errors and effectively accomplished both tracking tasks. We also tested the proposed fuzzy-model-based controller with our planar-spring-based continuum manipulator and the real-time tip position was recorded by the popular NDI Aurora® tracking system. The experimental results further confirmed the feasibility and superiority of the controller. In terms of tracking precision, the proposed fuzzy controller reached the target with sub-millimetre accuracy (the capstone result was 0.7165mm with our 3D printed manipulator). To the best of our knowledge, this is the first work of achieving task space closed-loop control in the field of continuum robotics using a fuzzy-model-based approach and the supervisor performances were successfully verified by both simulations and experiments.

6.2 Future work

Future research could proceed in the following directions, using the work in this thesis as a foundation.

6.2.1 Design of continuum manipulators with optimised configurations of planar springs

A further investigation of different design variations of planar springs would enable the optimisation of the proposed type of continuum manipulators for various application objectives. We have already noted that varying the angle of leg arrangement leads to the changing of its compliance properties as well as the outer space diameter of the planar spring. Thinking of the flexible segment – U-shape leg – in our current prototype, it would be interesting to provide more options of beams, for example, the curved beam (please refer to Figure 6-1). Apparently, the curved-leg design of the planar spring presented in Figure 6-1 results

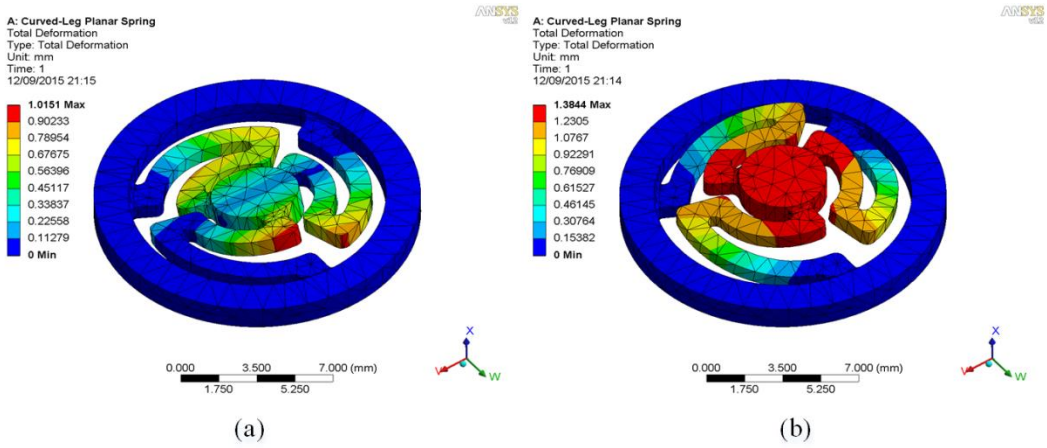


Figure 6-1 FEA simulations of the curved-leg planar spring using the commercial software package ANSYS®. The outer diameter of this planar spring design is 15mm. (a) a moment of 7N·mm about one axis in the layer plane was applied; (b) a force of 3N along the perpendicular axis to the layer plane was applied.

in a more compact structure. It would be crucial to establish the compliance model of this curved-leg planar spring and to investigate its compliance properties. The bottom-up approach employed in this thesis for deriving the compliance matrix of the standard rectangular beam might be applicable for this curved spring, but the seminal compliance matrix with respect to the curved beam would be complicated.

Besides, design optimisation for the current planar spring design would be very important. In order to apply this continuum manipulator for the practical use, the super-elastic metal fabrication will be ideal to maintain the mechanical properties. The design variables embodied in the developed compliance matrix could be optimised to achieve the best performance under certain constraints and complying with the properties of fabrication materials.

Multi-section designs of continuum manipulators could be investigated, whilst integrating different design variations of planar springs along the backbone. In view of different configurations of planar springs exhibiting different compliance properties, it would be possible to arrange different planar springs along the back to achieve variable-flexibility continuum manipulators with more complex deformations.

6.2.2 Integrating the intrinsic force sensing module via planar springs

This type of planar springs has been utilised to deliver force sensing with the optical sensing technique (Ataollahi, Fallah, Seneviratne, Dasgupta, & Althoefer, 2014). Now given a detailed analysis on both the translational and rotational deflections of planar springs, it would be possible to develop a force/toque

(F/T) sensor with this type of planar springs. Also, integrating such F/T sensing modules into the proposed continuum manipulator would lead to the union of both producing and detecting bending and translational motions. That is to say, each layer of the planar spring along the backbone of the proposed continuum manipulator can not only act as an actuator to produce the translational and rotational motions, but also act as a sensor to detect the movements. Therefore, the shape information of the continuum manipulator can be available.

Besides, the tip F/T sensing is highly desirable for many applications that aim to exert forces to the environments or objects (Xu & Simaan, 2008), (Xu & Simaan, 2010), (Rucker & Webster, 2011). The further developed continuum manipulator with an F/T sensor on its last planar spring module would achieve both functions of simultaneously producing and detecting the applied tip force to the environments or objects.

6.2.3 Study on the behaviours of the continuum manipulator interacting with environments

Most research on the behaviours of the continuum manipulator is focusing on free-space operation. In this thesis, the force-deflection model towards active actuation has been studied using a compliance matrix approach. Efforts to derive the body contact model with environments are expected to help further understand the intrinsic passive compliance property of continuum manipulators. In (Yip & Camarillo, 2014), the authors studied motion control of the continuum manipulators in constrained environments. The proposed control scheme was simple but effective without using a model of the continuum manipulator. The real-time Jacobians of the continuum manipulator is needed to be estimated based on tendon tension in the control process. If we could know the body contact model, we expect that it would be much better to cope with the constrained environments. This could be achieved with our proposed continuum manipulator together with the possibility to integrate the intrinsic F/T sensing module as discussed in 6.2.2.

If the body contact model is derived, then the continuum manipulator could more dexterously act as an elephant trunk to perform the “whole arm manipulation” (Salisbury, 1987). This would boost revolutionary developments in the area of robotic manipulation.

Continuum manipulators theoretically have infinite degrees of freedom (DOFs), although these DOFs cannot be wholly actuated on a continuum manipulator. It is helpful to activate more DOFs when interacting with environments or grasping objects. An analysis on the redundancy in these situations would be highly desirable to manipulate different shapes of objects with a continuum manipulator.

6.2.4 Computationally-efficient dynamic modelling and control

Acquiring a dynamic model of the continuum manipulator would enable a more realistic physical-based understanding of manipulator's behaviours. However, the current dynamic models established for continuum manipulators based on the classical approaches parallel to these formulations for conventional rigid-link manipulators are have a significant issue of computational complexity. Finding an efficient way to develop the dynamic model or applicable approximate dynamic model is an active research problem. In view of the complexity of kinematic model of continuum manipulators, we have utilised fuzzy control approaches to achieve manipulator's motion control. In this process of fuzzy-model-based control, the kinematics was first linearized at several operating points, and then the fuzzy model was used to blend these local models and thus acquiring a presentation of the whole system. The dynamic models of continuum manipulators are much more complexed than the kinematic models (it can be implied by those complex matrices in Appendix C). It would be interesting to locally linearize the dynamic models and apply the fuzzy-model-based control method to achieve dynamic control.

Bibliography

Althoefer, K. (1997). *Neuro-fuzzy motion planning for robotic manipulators.* PhD Thesis, Department of Electronic and Electrical Engineering, King's College London.

Althoefer, K., & Bugmann, G. (1995). Planning and learning goal-directed sequences of robot arm movements. *International Conference on Artificial Neural Networks*, (pp. 449-454).

Althoefer, K., & Seneviratne, L. (1999). Learning in a neuro-fuzzy navigator for robotic manipulators. *6th International Conference on Neural Information Processing* (pp. 347-352). IEEE.

Althoefer, K., Krekelberg, B., Husmeier, D., & Seneviratne, L. (2001). Reinforcement learning in a rule-based navigator for robotic manipulators. *Neurocomputing, 37*(1-4), 51-70.

Althoefer, K., Seneviratne, L., Zavlangas, P., & Krekelberg, B. (1998). Fuzzy navigation for robotic manipulators. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6*(2), 179-188.

Althofer, K., & Fraser, D. (1996). Fuzzy obstacle avoidance for robotic manipulators. *Neural Network World, 6*(2), 133-142.

Anderson, V. C., & Horn, R. C. (1967). Tensor arm manipulator design. *Transactions of the ASME, 67*(DE-57), 1-12.

Ataollahi, A., Fallah, A., Seneviratne, L., Dasgupta, P., & Althoefer, K. (2014). Novel force sensing approach employing prismatic-tip optical fiber inside an orthoplanar spring structure. *IEEE/ASME Transactions on Mechatronics, 19*(1), 121-130.

Ayvali, E., & Desai, J. P. (2012). Towards a discretely actuated steerable cannula. *IEEE International Conference on Robotics and Automation* (pp. 1614-1619). Saint Paul: IEEE.

Baer, R. (2005). *Linear Algebra and Projective Geometry.* Dover Publications.

Bailly, Y., Amirat, Y., & Fried, G. (2011). Modeling and control of a continuum style microrobot for endovascular surgery. *IEEE Transactions on Robotics, 27*(5), 1024-1030.

Ball, R. S. (1900). *A Treatise on the Theory of Screws.* Cambridge Univ. Press.

Beatty, M. F. (1986). *Principles of Engineering Mechanics: Volume 1: Kinematics.* New York: Plenum Press.

Braganza, D., Dawson, D. M., Walker, I. D., & Nath, N. (2007). A neural network controller for continuum robots. *IEEE Transactions on Robotics, 23*(6), 1270-1277.

Brand, L. (1949). *Vector and Tensor Analysis.* New York: John Wiley & Sons.

Buckingham, R. (2002). Snake arm robots. *Industrial Robot, 29*(3), 242-245.

Burgner, J., Rucker, D. C., Gilbert, H. B., Swaney, P. J., Russell, P. T., Weaver, K. D., & Webster, R. J. (2014). A telerobotic system for transnasal surgery. *IEEE/ASME Transaction on Mechatronics, 19*(3), 996-1006.

Camarillo, D. B., Krummel, T. M., & Salisbury, J. K. (2004). Robotic technology in surgery: past, present, and future. *The American Journal of Surgery, 188*(4A), 2-15.

Camarillo, D. B., Milne, C. F., Carlson, C. R., Zinn, M. R., & Salisbury, J. K. (2008). Mechanics modeling of tendon-driven continuum manipulators. *IEEE Transactions on Robotics, 24*(6), 1262-1273.

Chirikjian, G. S. (1994). Hyper-redundant manipulator dynamics: a continuum approximation. *Advanced Robotics, 9*(3), 217-243.

Chitrakaran, V. K., Behal, A., Dawson, D. M., & Walker, I. D. (2007). Setpoint regulation of continuum robots using a fixed camera. *Robotica, 25*(05), 581-586.

Cianchetti, M., Ranzani, T., Gerboni, G., Falco, I. D., Laschi, C., & Menciassi, A. (2013). STIFF-FLOP surgical manipulator: mechanical design and experimental characterization of the single module. *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3576-3581). Tokyo: IEEE.

Craig, J. J. (2005). *Introduction to robotics: mechanics and control.* Prentice Hall.

Dai, J. S. (2006). An historical review of the theoretical development of rigid body displacements from Rodrigues parameters to the finite twist. *Mechanism and Machine Theory, 41*(1), 41-52.

Dai, J. S. (2012). Finite displacement screw operators with embedded Chasles' motion. *Journal of Mechanisms and Robotics, 4*(4), 041002.

Dai, J. S. (2014). *Geometrical Foundations and Screw Algebra for Mechanisms and Robotics.* Beijing: Higher Education Press.

Dai, J. S. (2015). Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections. *Mechanism and Machine Theory, 92*, 144-152.

Dai, J. S., Holland, N., & Kerr, D. R. (1995). Finite twist mapping and its application to planar serial manipulators with revolute joints. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 209*(4), 263-271.

Dupont, P. E., Lock, J., Itkowitz, B., & Butler, E. (2010). Design and control of concentric-tube robots. *IEEE Transactions on Robotics, 26*(2), 209-225.

Escande, C., Chettibi, T., Merzouki, R., Coelen, V., & Pathak, P. M. (2015). Kinematic calibration of a multisection bionic manipulator. *IEEE/ASME Transactions on Mechatronics, 20*(2), 663-674.

Festo Corporate. (2010). *Bionic Handling Assistant – flexible and compliant movement*. Retrieved from http://www.festo.com/cms/en_corp/9655_10218.htm

Girard, A., Bigue, J. P., O'Brien, B. M., Gisby, T. A., Anderson, I. A., & Plante, J. S. (2015). Soft two-degree-of-freedom dielectric elastomer position sensor exhibiting linear behavior. *IEEE/ASME Transactions on Mechatronics, 20*(1), 105-114.

Giri, N., & Walker, I. D. (2011). Three module lumped element model of a continuum arm section. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 4060-4065).

Gravagne, I. A., Rahn, C. D., & Walker, I. D. (2003). Large deflection dynamics and control for planar continuum robots. *IEEE/ASME Transactions on Mechatronics, 8*(2), 299-307.

Hannan, M. A., & Walker, I. D. (2003). Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots. *Journal of Robotic Systems, 20*(2), 45-63.

Hansen Medical Inc. (n.d.). Retrieved from http://www.hansenmedical.com/

Helgason, S. (1979). *Differential Geometry, Lie Groups, and Symmetric Spaces (Vol. 80).* Academic Press.

Hirose, S. (1993). *Biologically Inspired Robots.* Oxford University Press.

Howell, L. L. (2001). *Compliant Mechanisms.* John Wiley & Sons, Inc.

Howell, L. L., Thomson, S., Briscoe, J. A., Parise, J. J., Lorenc, S., Larsen, J. B., . . . Gomm, T. A. (2000). *USA Patent No. WO2001081785 A1, US20040021123 A1, US6983924 B2.*

Huang, C., & Roth, B. (1994). Analytic expressions for the finite screw systems. *Mechanism and Machine Theory, 29*(2), 207-222.

Hunt, K. H. (1990). *Kinematic Geometry of Mechanisms.* Oxford University Press .

Ivanescu, M. (2002). Position dynamic control for a tentacle manipulator. *IEEE International Conference on Robotics and Automation*, (pp. 1531-1538). Washington, DC.

Jel ńek, F., Arkenbout, E. A., Henselmans, P. W., Pessers, R., & Breedveld, P. (2015). Classification of joints used in steerable instruments for minimally invasive surgery. *ASME Journal of Medical Devices, 8*(3), 030914.

Jiang, A., Xynogalas, G., Dasgupta, P., Althoefer, K., & Nanayakkara, T. (2012). Design of a variable stiffness flexible manipulator with composite granular jamming and membrane coupling. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 2922-2927).

Jung, J., Penning, R. S., Ferrier, N. J., & Zinn, M. R. (2011). A modeling approach for continuum robotic manipulators: effects of nonlinear internal device friction. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 5139-5146).

Kanno, T., Haraguchi, D., Yamamoto, M., Tadano, K., & Kawashima, K. (2015). A forceps manipulator with flexible 4-DOF mechanism for laparoscopic surgery. *IEEE/ASME Transactions on Mechatronics, 20*(3), 1170-1178.

Kapadia, A., & Walker, I. D. (2011). Task-space control of extensible continuum manipulators. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 1087-1092). San Francisco.

Kesner, S. B., & Howe, R. D. (2011). Position control of motion compensation cardiac catheters. *IEEE Transactions on Robotics, 27*(6), 1045-1055.

Kier, W. M., & Smith, K. K. (1985). Tongues, tentacles and trunks: the biomechanics of movement in muscular-hydrostats. *Zoological Journal of the Linnean Society, 83*(4), 307-324.

Kim, Y. J., Cheng, S., Kim, S., & Iagnemma, K. (2013). A novel layer jamming mechanism with tunable stiffness capability for minimally invasive surgery. *IEEE Transactions on Robotics, 29*(4), 1031-1042.

Klein, F. (1872). *Vergleichende betrachtungen über neuere geometrische forsuchungen.* A. Deichert.

Koolwal, A., Barbagli, F., Carlson, C., & Liang, D. (2010). An ultrasound-based localization algorithm for catheter ablation guidance in the left atrium. *The International Journal of Robotics Research, 29*(6), 643-665.

Kumar, V. (n.d.). *MEAM 520 notes: The theorems of Euler and Chasles.* Retrieved from http://www.seas.upenn.edu/~meam520/notes02/EulerChasles4.pdf

Lam, H. K., & Leung, F. H. (2011). *Stability Analysis of Fuzzy-Model-Based Control Systems: Linear-Matrix-Inequality Approach.* Springer.

Lam, H. K., & Li, H. (2013). Output-feedback tracking control for polynomial fuzzy-model-based control systems. *IEEE Transactions on Industrial Electronics, 60*(12), 5830-5838.

Lee, T. H., Lam, H. K., Leung, F. H., & Tam, P. K. (2001). A fast path planning-and-tracking control for wheeled mobile robots. *IEEE International Conference on Robotics and Automation* (pp. 1736-1741). Seoul: IEEE.

Lee, T. H., Lam, H. K., Leung, F. H., & Tam, P. K. (2003). A practical fuzzy logic controller for the path tracking of wheeled mobile robots. *IEEE Control Systems Magazine, 23*(2), 60-65.

Liao, B., Li, Z., & Du, R. (2012). Robot tadpole with a novel biomimetic wire-driven propulsor. *IEEE International Conference on Robotics and Biomimetics* (pp. 557–562). Guangzhou: IEEE.

Mahl, T., Hildebrandt, A., & Sawodny, O. (2014). A variable curvature continuum kinematics for kinematic control of the bionic handling assistant. *IEEE Transactions on Robotics, 30*(4), 935-949.

Mahl, T., Mayer, A. E., Hildebrandt, A., & Sawodny, O. (2013). A variable curvature modeling approach for kinematic control of continuum manipulators. *American Control Conference*, (pp. 4945-4950). Washington, DC.

McMahan, W., Chitrakaran, V., Csencsits, M., Dawson, D., Walker, I. D., Jones, B. A., . . . Rahn, C. D. (2006). Field trials and testing of the OctArm continuum manipulator. *IEEE International Conference on Robotics and Automation* (pp. 2336-2341). Orlando, FL: IEEE.

McMahan, W., Jones, B. A., & Walker, I. D. (2005). Design and implementation of a multi-section continuum robot: Air-Octor. *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2578-2585). IEEE.

Mehling, J. S., Diftler, M., Chu, M., & Valvo, M. (2006). A minimally invasive tendril robot for in-space inspection. *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, (pp. 690- 695).

Melingui, A., Lakhal, O., Daachi, B., Mbede, J. B., & Merzouki, R. (2015). Adaptive neural network control of a compact bionic handling arm. *IEEE/ASME Trans. Mechatronics, 99*, 1-14.

Mochiyama, H., & Suzuki, T. (2002). Dynamical modelling of a hyperflexible manipulator. *The 41st SICE Annual Conference*, *3*, pp. 1505-1510.

Mochiyama, H., & Suzuki, T. (2003). "Kinematics and dynamics of a cablelike hyper-flexible manipulator. *IEEE International Conference on Robotics and Automation*, (pp. 3672-3677).

Murray, R., Li, Z., & Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation.* CRC Press.

Parise, J. J., Howell, L. L., & Magleby, S. P. (2001). Orthoplanar linear-motion springs. *Mechanism and Machine Theory, 36*(11), 1281-1299.

Penning, R. S., & Zinn, M. (2014). A combined modal-joint space control approach for continuum manipulators. *Advanced Robotics, 28*(16), 1091-1108.

Penning, R. S., Jung, J., Borgstadt, J. A., Ferrier, N. J., & Zinn, M. R. (2011). Towards closed loop control of a continuum robotic manipulator for medical applications. *IEEE International Conference on Robotics and Automation*, (pp. 4822-4827). Shanghai.

Penning, R. S., Jung, J., Ferrier, N. J., & Zinn, M. R. (2012). An evaluation of closed-loop control options for continuum manipulators. *IEEE International Conference on Robotics and Automation*, (pp. 5392-5397). Minnesota.

Prajna, S., Papachristodoulou, A., & Parrilo, P. A. (2002). SOSTOOLS—Sum of squares optimization toolbox, user's guide. Natick, MA.

Pritts, M. B., & Rahn, C. D. (2004). Design of an artificial muscle continuum robot. *IEEE International Conference on Robotics and Automation*, (pp. 4742-4746).

Roberts, R. G. (1999). Minimal realization of a spatial stiffness matrix with simple springs connected in parallel. *IEEE Transactions on Robotics and Automation, 15*(5), 953-958.

Roberts, R. G. (2000). Minimal realization of an arbitrary spatial stiffness matrix with a parallel connection of simple and complex springs. *IEEE Transactions on Robotics and Automation, 16*(5), 603-608.

Robinson, G., & Davies, J. B. (1999). Continuum robots—a state of the art. *IEEE International Conference on Robotics and Automation* (pp. 2849-2854). Detroit: IEEE.

Rucker, D. C., & Webster, R. J. (2011). Deflection-based force sensing for continuum robots: A probabilistic approach. *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3764-3769). San Francisco: IEEE.

Rucker, D. C., & Webster, R. J. (2011). Statics and dynamics of continuum robots with general tendon routing and external loading. *IEEE Transactions on Robotics, 27*(6), 1033-1044.

Salisbury, J. K. (1987). Whole arm manipulation. *4th Symposium on Robotics Research.* MIT Press.

Selig, J. M. (2004). *Geometric Fundamentals of Robotics.* Springer.

Selig, J., & Ding, X. (2001). A screw theory of static beams. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 312-317).

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics: modelling, planning and control.* London: Springer-Verlag.

Simaan, N. (2005). Snake-like units using flexible backbones and actuation redundancy for enhanced miniaturization. *IEEE Internationa lConference on Robotics and Automation*, (pp. 3012-3017).

Simaan, N., Xu, K., Kapoor, A., Wei, W., Kazanzides, P., Flint, P., & Taylor, R. H. (2009). Design and integration of a telerobotic system for minimally invasive surgery of the throat. *The International Journal of Robotics Research, 28*(9), 1134-1153.

Spivak, M. (1999). *A Comprehensive Introduction to Differential Geometry.* Publish or Perish, Inc.

Stancin, S., & Tomazic, S. (2011). Angle estimation of simultaneous orthogonal rotations from 3D gyroscope measurements. *Sensors, 11*(9), 8536-8549.

Su, H. J., Shi, H., & Yu, J. (2012). A symbolic formulation for analytical compliance analysis and synthesis of flexure mechanisms. *ASME Journal of Mechanical Design, 134*, 051009.

Tanaka, K., & Wang, H. O. (2004). *Fuzzy control systems design and analysis: a linear matrix inequality approach.* John Wiley & Sons.

Tanaka, K., Yoshida, H., Ohtake, H., & Wang, H. O. (2009). A sum-of-squares approach to modeling and control of nonlinear dynamical systems with polynomial fuzzy systems. *IEEE Transactions on Fuzzy Systems, 17*(4), 911-922.

Tatlicioglu, E., Walker, I. D., & Dawson, D. M. (2007). Dynamic modelling for planar extensible continuum robot manipulators. *IEEE International Conference on Robotics and Automation,* (pp. 1357-1362).

Trivedi, D., Rahn, C. D., Kier, W. M., & Walker, I. D. (2008). Soft robotics: biological inspiration, state of the art, and future research. *Applied Bionics and Biomechanics, 5*(3), 99-117.

Tsai, L. W. (1999). *Robot Analysis: The Mechanics of Serial and Parallel Manipulators.* Wiley-Interscience.

von Mises, R. M. (1924). ein neues hilfsmittel in der mechanik. *Zeitschrift für Angewandte Mathematik und Mechanik, 4*(2), 155-181.

Walker, D. (2013). Continuous backbone “continuum” robot manipulators. *ISRN Robotics,* Article ID 726506.

Wang, H. O., Tanaka, K., & Griffin, M. F. (1996). An approach to fuzzy control of nonlinear systems: stability and design issues. *IEEE Transactions on Fuzzy Systems, 4*(1), 14-23.

Watanabe, H., Kanou, K., Kobayashi, Y., & Fujie, M. G. (2011). Development of a “steerable drill" for ACL reconstruction to create the arbitrary trajectory of a bone tunnel. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 955-960).

Watanabe, H., Kanou, K., Kobayashi, Y., & Fujie, M. G. (2011). Development of a “steerable drill” for ACL reconstruction to create the arbitrary trajectory of a bone tunnel. *IEEE/RSJ International Conference on Intelligent Robots and Systems,* (pp. 955-960).

Webster, R. J., & Jones, B. A. (2010). Design and kinematic modeling of constant curvature continuum robots: a review. *The International Journal of Robotics Research, 29*(13), 1661-1683.

Webster, R. J., Joseph, R., & Cowan, N. J. (2009). Mechanics of precurved-tube continuum robots. *IEEE Transactions on Robotics, 25*(1), 67-78.

Wright, C., Johnson, A., Peck, A., McCord, Z., Naaktgeboren, A., Gianforton, P., . . . Choset, H. (2007). Design of a modular snake robot. *IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2609-2614). San Diego: IEEE.

Xu, K., & Simaan, N. (2008). An investigation of the intrinsic force sensing capabilities of continuum robots. *IEEE Transactions on Robotics, 24*(3), 576-587.

Xu, K., & Simaan, N. (2010). Analytic formulation for kinematics, statics, and shape restoration of multibackbone continuum robots via elliptic integrals. *Journal of Mechanisms and Robotics, 2*(1), 011006.

Xu, K., & Simaan, N. (2010). Intrinsic wrench estimation and its performance index of multi-segment continuum robots. *IEEE Transactions on Robotics, 26*(3), 555-561.

Yang, J., Jason, P., & Abdel-Malek, K. (2006). A hyper-redundant continuous robot. *IEEE International Conference on Robotics and Automation* (pp. 1854-1859). IEEE.

Yip, M. C., & Camarillo, D. B. (2014). Model-less feedback control of continuum manipulators in constrained environments. *IEEE Transactions on Robotics, 30*(4), 880-889.

Yuen, S. G., Kettler, D. T., Novotny, P. M., Plowes, R. D., & Howe, R. D. (2009). Robotic motion compensation for beating heart intracardiac surgery. *The International Journal of Robotics Research, 28*(10), 1355-1372.

Zhang, L., khare, R., Wilson, E., Wang, S., Peters, C. A., & Cleary, K. (2014). Robotic assistance for manipulating a flexible endoscope. *IEEE International Conference on Robotics and Automation*, (pp. 5380-5385).

Appendix A. Continuum Manipulator Assembly with Motors

The continuum manipulator is built with multiple double-layer modules. The current prototype designed for 3D printing is illustrated as below. For simplicity, three DC motors are directly implemented with a pulley to pull or release the tendons.

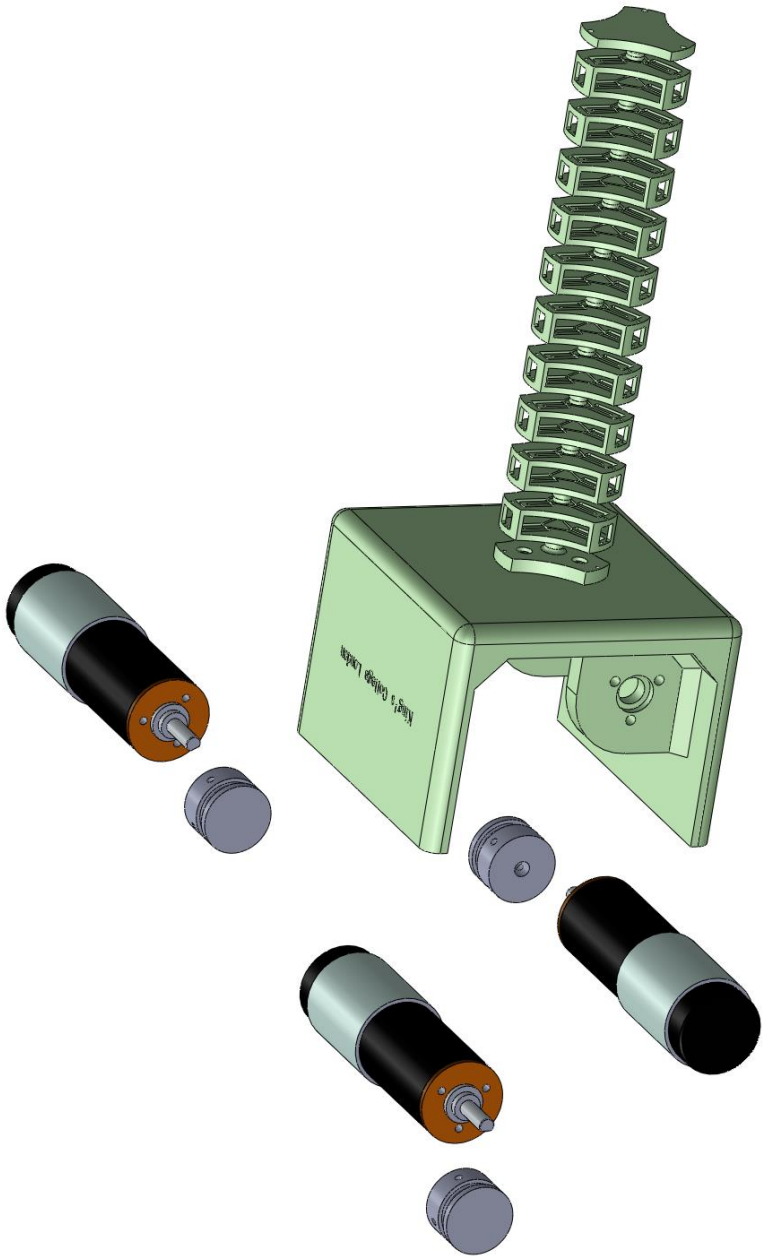


Figure A-1 Continuum manipulator assembly.

Appendix B. Design Variations of the Planar Spring

Design variations of the U-shape leg planar spring with varying angle of leg arrangement are depicted as below. The configuration starts from the standard side-leg design, and then increases the angle α in an interval of 10˚. It can be seen that with the angle increasing the diameter of the planar spring contour becomes larger and larger.

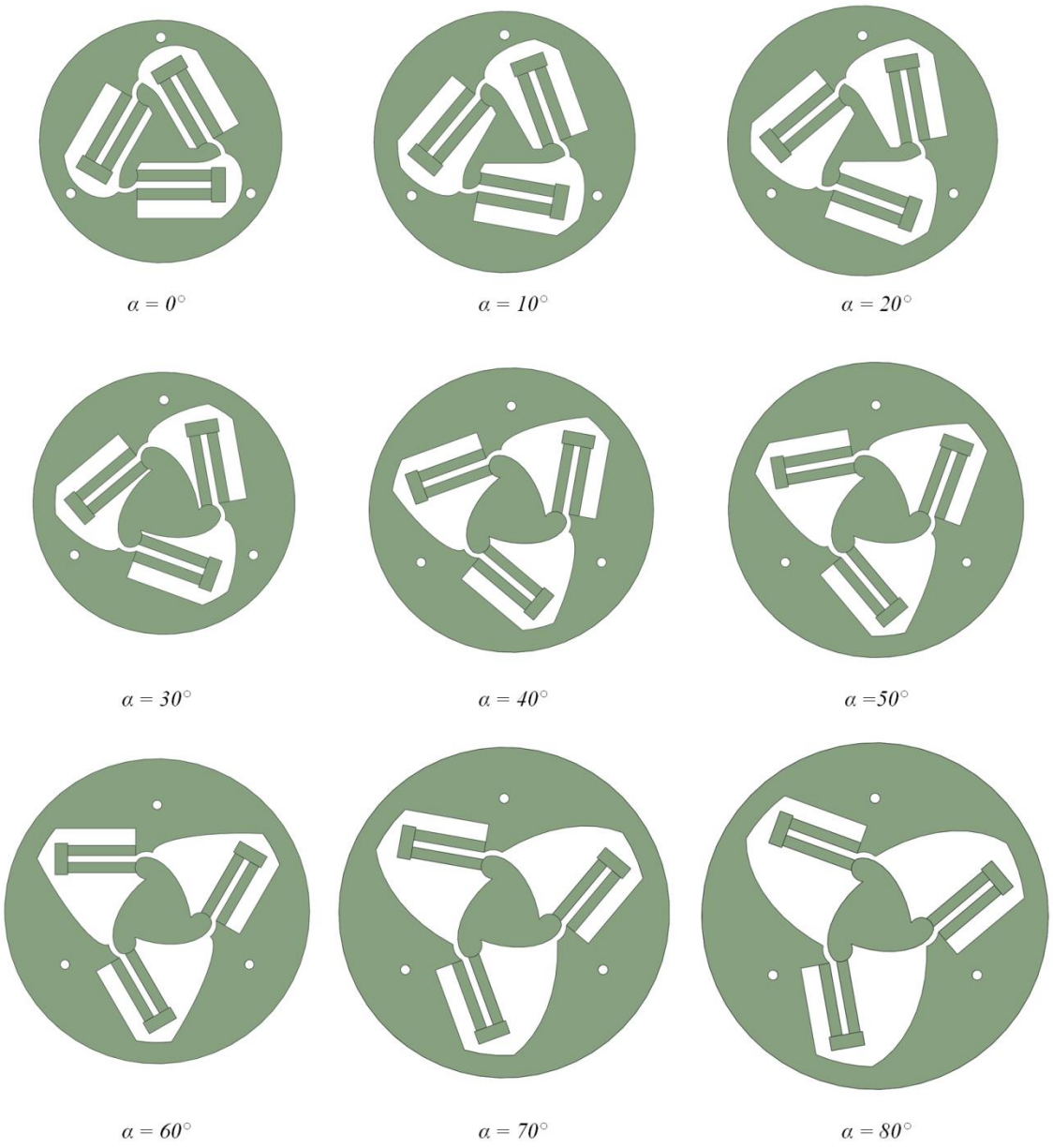


Figure B-1 Design variations of the planar spring. The angle of leg arrangement is varying from 0˚ to 80˚ with an interval of 10˚.

Appendix C. The Entries of Matrices in Dynamic Models

The dynamic model of the continuum manipulator has been derived in the following form,

$$\mathbf{M}(q)\ddot{q}+\mathbf{V}(q,\dot{q})\dot{q}+\mathbf{G}(q)=\tau$$

where $\mathbf{M}(q)\in\mathbb{R}^{2\times2}$ is the manipulator inertia matrix, $\mathbf{V}(q,\dot{q})\in\mathbb{R}^{2\times2}$ is the matrix of centrifugal Coriolis torque and $\mathbf{G}(q)\in\mathbb{R}^{2}$ is the matrix about gravitational torques; the configuration-space variables are presented as $q = [l, k]^T$, and the first and second order derivatives are denoted as $\dot{q},\ddot{q}$. The constant parameters in the derivation are set be as in Table C-1. Subsequently, the entries of the matrices $\mathbf{M}(q)$, $\mathbf{V}(q,\dot{q})$, and $\mathbf{G}(q)$ are as follows.

Table C-1 The constant parameter setting in the derivation of dynamic models of a continuum manipulator.

Name and notation	***Value with unit***
*Relaxed length d**	0.1m
Gravitational acceleration g	9.8N/kg
Cross-section radius r	0.01m
Mass m	0.05kg
Bending spring constant k_b	0.001
Elastic spring constant k_e	10

$$\mathbf{M}(q)=\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

$$M_{11}=0.1l^2k^3-0.1lk^2\sin(lk)$$

$$M_{12}=l^2k^2\left(0.05(\cos(lk))^2-0.15\cos(lk)\right)+lk\sin(lk)(0.25-0.25\cos(lk))-0.125l^2k^2$$

$$\begin{aligned} M_{21}=\left(l^2k^6\right)^{-1}\big(&-0.125l^2k^3-0.025lk^2\sin(lk)\cos(lk)\ldots \\ &+0.05l^2k^3(\cos(lk))^2-0.15l^2k^3\cos(lk)+0.25lk^2\sin(lk)\big)\end{aligned}$$

$$\begin{aligned} M_{22}=\left(l^2k^6\right)^{-1}\big(&0.05l^4k^4+0.2l^2k^2+0.1lk\sin(lk)\cos(lk)-0.1l^3k^3\sin(lk)\cos(lk)\ldots \\ &-0.4lk\sin(lk)+0.3l^2k^2\cos(lk)-0.2l^2k^2(\cos(lk))^2\big)\end{aligned}$$

$$\mathbf{V}(q,\dot{q}) = \begin{bmatrix} V_{11}+W_{11} & V_{12}+W_{12} \\ V_{21}+W_{21} & V_{22}+W_{22} \end{bmatrix}$$

$$\begin{aligned} V_{11} = \dot{k}\Big(&-0.1l^2k^3\sin(lk)\cos(lk)+0.3lk\sin(lk)-0.1l^2k^2\cos(lk)+0.025k\sin(lk)\cos(lk)\ldots \\ &-0.05lk^2\left(\cos(lk)\right)^2+0.25lk^2\cos(lk)+0.15l^2k^3\sin(lk)+0.025lk^2-0.2l^2k^2-0.25k\sin(lk)\Big)\ldots \\ &+\dot{l}\left(0.1k^2\sin(lk)-0.1lk^3\cos(lk)\right) \end{aligned}$$

$$\begin{aligned} V_{12} = \dot{k}\Big(&-0.1l^3k^2\sin(lk)\cos(lk)+0.4l^2k-l\sin(lk)-0.2l^2k\left(\cos(lk)\right)^2\ldots \\ &+0.1l\sin(lk)\cos(lk)+0.7l^2k\cos(lk)+0.15l^3k^2\sin(lk)\Big) \end{aligned}$$

$$\begin{aligned} V_{21} = \dot{l}\left(l^2k^6\right)^{-1}\Big(&-0.1l^2k^4\sin(lk)\cos(lk)-0.25k^2\sin(lk)+0.15l^2k^4\sin(lk)\ldots \\ &+0.25lk^3\cos(lk)-0.05lk^3\left(\cos(lk)\right)^2+0.025k^2\sin(lk)\cos(lk)\Big) \end{aligned}$$

$$\begin{aligned} V_{22} = \dot{l}\left(l^2k^6\right)^{-1}\Big(&-0.1lk^2+0.3l^2k^3\sin(lk)\cos(lk)+0.1lk\sin(lk)\cos(lk)\ldots \\ &-0.1l^3k^3\sin(lk)\cos(lk)-0.2l^3k^4\left(\cos(lk)\right)^2-lk\sin(lk)-0.3l^2k^3\sin(lk)\ldots \\ &-0.4lk^2\cos(lk)+0.2lk^2\left(\cos(lk)\right)^2-0.1k\sin(lk)\cos(lk)+0.15l^3k^3\sin(lk)\ldots \\ &+0.7l^2k^2\cos(lk)-0.2l^2k^2\left(\cos(lk)\right)^2+0.4k\sin(lk)+0.2l^3k^4+0.4l^2k^2\Big)\ldots \\ &+\dot{k}\left(l^2k^6\right)^{-1}\Big(0.7l^3k^2\sin(lk)\cos(lk)-0.9l^2k+2l\sin(lk)-0.2l^4k^3\left(\cos(lk)\right)^2\ldots \\ &-0.3l^3k^2\sin(lk)-1.6l^2k\cos(lk)+l^2k\left(\cos(lk)\right)^2-0.5l\sin(lk)\cos(lk)\Big) \end{aligned}$$

$$\begin{aligned} W_{11} = \dot{k}\left(l^2k^5\right)^{-1}\Big(&0.1l^2k^3\sin(lk)\cos(lk)-0.025k\sin(lk)\cos(lk)-0.05lk^2\left(\cos(lk)\right)^2\ldots \\ &-0.25lk^2\cos(lk)-0.15l^2k^3\sin(lk)-0.025lk^2+0.25k\sin(lk)\Big)\ldots \\ &+\dot{l}\left(l^2k^5\right)^{-1}\left(-0.05k^2\sin(lk)+0.05lk^3\cos(lk)\right) \end{aligned}$$

$$\begin{aligned} W_{12} = \dot{k}\left(l^2k^5\right)^{-1}\Big(&-0.15l^2k^2\sin(lk)\cos(lk)-0.2\sin(lk)+0.05\sin(lk)\cos(lk)-0.1l^3k^3\ldots \\ &-0.1lk\left(\cos(lk)\right)^2+0.2lk\cos(lk)+0.15l^2k^2\sin(lk)+0.1l^3k^3\left(\cos(lk)\right)^2+0.05lk\Big) \end{aligned}$$

$$W_{21} = \dot{l}\left(lk^6\right)^{-1}\left(-0.1lk^3-0.15k^2\sin(lk)+0.05lk^3\cos(lk)\right)$$

$$\begin{aligned} W_{22} = \dot{l}\left(lk^6\right)^{-1}\Big(&-0.1\sin(lk)\cos(lk)-0.15l^2k^3\sin(lk)+0.2lk^2\left(\cos(lk)\right)^2\ldots \\ &-0.7lk^2\cos(lk)-0.4lk^2+k\sin(lk)+0.1l^2k^3\sin(lk)\cos(lk)\Big)\ldots \\ &+\dot{k}\left(lk^6\right)^{-1}\Big(-0.35l^2k^2\sin(lk)\cos(lk)+0.25\sin(lk)\cos(lk)+0.45lk+0.15l^2k^2\sin(lk)\ldots \\ &-0.5lk\left(\cos(lk)\right)^2+0.8lk\cos(lk)+0.1l^3k^3\left(\cos(lk)\right)^2-\sin(lk)\Big) \end{aligned}$$

$$\mathbf{G}(q) = \begin{bmatrix} G_{11} \\ G_{21} \end{bmatrix}$$

$$G_{11} = \left(l^2 k^5\right)^{-1} \left(0.49k^3 + l^2 k^5 - 10l^3 k^5 + 0.49k^3 \cos(lk) + 0.49lk^4 \sin(lk)\right)$$

$$G_{21} = \left(lk^6\right)^{-1} \left(0.98k^3 + 0.49lk^4 \sin(lk) + 0.98k^3 \cos(lk)\right)$$

Appendix D. Straight Line Tracking Simulation and Control Code

1) fbf_cs.m

```
% Control signal (closed-loop Jacobian-based control)
function u = fbf_cs(t,y, X0, Y0, Z0, X_ref, Y_ref, Z_ref, method)
count = size(t,1);
u = zeros(count,3);   % u represents control signal.
for ii = 1:count
    tt = t(ii,1);

    x1 = y(ii,1);
    x2 = y(ii,2);
    x3 = y(ii,3);
    xx = [x1; x2; x3];

    l1 = y(ii,4);
    l2 = y(ii,5);
    l3 = y(ii,6);

    xr1 = y(ii,7);
    xr2 = y(ii,8);
    xr3 = y(ii,9);
    xr = [xr1; xr2; xr3];

    d = 0.005; % the radius of the cross-section of the continuum manipulator; unit: meter.

    % to define the reference model.
    Ar = zeros(3,3);
    Br = 1/60*[X_ref - X0; Y_ref - Y0; Z_ref - Z0];

    if tt <= 60     % after 60 (seconds), the moving reference target will stop.
```

```
        rt = 1;
    else
        rt = 0;
    end

    xr_dot = Ar*xr + Br*rt;     % To generate the reference trajectory.
    % Elements of the 3×3 Jacobian matrix calculated as below, which depend on the three lengths of three tendons and the cross-section radius of the continuum manipulator.
    J11 = Jelement_11(l1, l2, l3, d);
    J12 = Jelement_12(l1, l2, l3, d);
    J13 = Jelement_13(l1, l2, l3, d);
    J21 = Jelement_21(l1, l2, l3, d);
    J22 = Jelement_22(l1, l2, l3, d);
    J23 = Jelement_23(l1, l2, l3, d);
    J31 = Jelement_31(l1, l2, l3, d);
    J32 = Jelement_32(l1, l2, l3, d);
    J33 = Jelement_33(l1, l2, l3, d);
    % 3×3 Jacobian matrix
    J = [J11, J12,  J13;
        J21, J22,  J23;
        J31, J32,  J33];
    % Controller gain
    K = 0.1*eye(3);
    u(ii,:) = (inv(J)*(xr_dot + K*(xr - xx)))';        % control signal for controlling the three tendon's speeds.
end
```

2) fbf_ode.m

```
% Ordinary differential equations
function dx = fbf_ode(t, x, X0, Y0, Z0, X_ref, Y_ref, Z_ref, method)
    x1 = x(1);
    x2 = x(2);
    x3 = x(3);
```

```
xx = [x1; x2; x3];


l1 = x(4);

l2 = x(5);

l3 = x(6);


xr1 = x(7);

xr2 = x(8);

xr3 = x(9);

xr = [xr1; xr2; xr3];


d = 0.005; % the radius of the cross-section of the continuum manipulator; unit: meter.


% to define the reference model.

Ar = zeros(3,3);

Br = 1/60*[X_ref - X0; Y_ref - Y0; Z_ref - Z0];


if t <= 60     % after 60 (seconds), the moving reference target will stop.

   rt = 1;

else

   rt = 0;

end


xr_dot = Ar*xr + Br*rt;     % To generate the reference trajectory.

 % Elements of the 3×3 Jacobian matrix calculated as below, which depend on the three lengths of three tendons and the cross-section radius of the continuum manipulator.

J11 = Jelement_11(l1, l2, l3, d);

J12 = Jelement_12(l1, l2, l3, d);

J13 = Jelement_13(l1, l2, l3, d);

J21 = Jelement_21(l1, l2, l3, d);

J22 = Jelement_22(l1, l2, l3, d);

J23 = Jelement_23(l1, l2, l3, d);
```

```
J31 = Jelement_31(l1, l2, l3, d);
J32 = Jelement_32(l1, l2, l3, d);
J33 = Jelement_33(l1, l2, l3, d);
 % 3×3 Jacobian matrix
J = [J11, J12,  J13;
   J21, J22,  J23;
   J31, J32,  J33];
K = 0.1*eye(3);     % Controller gain
 u = inv(J)*(xr_dot + K*(xr - xx));       % control signal for controlling the three tendon's speeds.
```

% the real continuum manipulator plant is simulated in our simulations when testing the controllers. The physical meaning of this Delta J (0.2*ones(3,3), as below) is to represent the system error, such as frictions, hardware defects or fatigue, etc. In our case, we added Delta J with the value 0.2 as the additive disturbance term. In real situation, many factors will lead to disturbance and these errors are within a certain range. We have tested different values to simulate those disturbances, and 0.2 is approximately the mean value to simulate disturbance in our experiment setup.

```
J = J + 0.2*ones(3,3);

dx(1:3,:) = J*u;
dx(4:6,:) = u(1:3);
dx(7:9,:) = xr_dot;
```

3) ff_cs.m

```
% Control signal (Open-loop feed-forward control)
function u = ff_cs(t,y, X0, Y0, Z0, X_ref, Y_ref, Z_ref,method)
count = size(t,1);
u = zeros(count,3);
for ii = 1:count

tt = t(ii,1);

x1 = y(ii,1);
```

```
x2 = y(ii,2);

x3 = y(ii,3);

xx = [x1; x2; x3];


l1 = y(ii,4);

l2 = y(ii,5);

l3 = y(ii,6);


xr1 = y(ii,7);

xr2 = y(ii,8);

xr3 = y(ii,9);

xr = [xr1; xr2; xr3];


d = 0.005; % the radius of the cross-section of the continuum manipulator; unit: meter.


% reference model

Ar = zeros(3,3);

Br = 1/60*[X_ref - X0; Y_ref - Y0; Z_ref - Z0];


if tt <= 60     % after 60 (seconds), the moving reference target will stop.

   rt = 1;

else

   rt = 0;

end


xr_dot = Ar*xr + Br*rt; ;     % To generate the reference trajectory.


% Elements of the 3×3 Jacobian matrix calculated as below, which depend on the three lengths of three tendons and the cross-section radius of the continuum manipulator.

J11 = Jelement_11(l1, l2, l3, d);

J12 = Jelement_12(l1, l2, l3, d);

J13 = Jelement_13(l1, l2, l3, d);
```

```
    J21 = Jelement_21(l1, l2, l3, d);
    J22 = Jelement_22(l1, l2, l3, d);
    J23 = Jelement_23(l1, l2, l3, d);
    J31 = Jelement_31(l1, l2, l3, d);
    J32 = Jelement_32(l1, l2, l3, d);
    J33 = Jelement_33(l1, l2, l3, d);

     % 3×3 Jacobian matrix
    J = [J11, J12,  J13;
        J21, J22,  J23;
        J31, J32,  J33];

    u(ii,:) = (inv(J)*xr_dot)';
end
```

4) ff_ode.m

```
% Odinary differential equations
function dx = ff_ode(t, x, X0, Y0, Z0, X_ref, Y_ref, Z_ref, method)

    x1 = x(1);
    x2 = x(2);
    x3 = x(3);
    xx = [x1; x2; x3];

    l1 = x(4);
    l2 = x(5);
    l3 = x(6);

    xr1 = x(7);
    xr2 = x(8);
    xr3 = x(9);
    xr = [xr1; xr2; xr3];
```

```
d = 0.005; % m

% reference model
Ar = zeros(3,3);
Br = 1/60*[X_ref - X0; Y_ref - Y0; Z_ref - Z0];

if t <= 60     % after 60 (seconds), the moving reference target will stop.
   rt = 1;
else
   rt = 0;
end

xr_dot = Ar*xr + Br*rt; ;     % To generate the reference trajectory.

% Elements of the 3×3 Jacobian matrix calculated as below, which depend on the three lengths of three tendons and the cross-section radius of the continuum manipulator.
J11 = Jelement_11(l1, l2, l3, d);
J12 = Jelement_12(l1, l2, l3, d);
J13 = Jelement_13(l1, l2, l3, d);
J21 = Jelement_21(l1, l2, l3, d);
J22 = Jelement_22(l1, l2, l3, d);
J23 = Jelement_23(l1, l2, l3, d);
J31 = Jelement_31(l1, l2, l3, d);
J32 = Jelement_32(l1, l2, l3, d);
J33 = Jelement_33(l1, l2, l3, d);

 % 3×3 Jacobian matrix
J = [J11, J12,  J13;
   J21, J22,  J23;
   J31, J32,  J33];
```

```
    u = inv(J)*xr_dot;

    J = J + 0.2*ones(3,3);

    dx(1:3,:) = J*u;
    dx(4:6,:) = u(1:3);
    dx(7:9,:) = xr_dot;
```

5) mnp_cs.m

```
% Control signal (fuzzy-model-based control)
function u = mnp_cs(t,y,method)

count = size(t,1);
u = zeros(count,3);

for ii = 1:count

    x1 = y(ii,1);
    x2 = y(ii,2);
    x3 = y(ii,3);
    xx = [x1; x2; x3];

    xr1 = y(ii,7);
    xr2 = y(ii,8);
    xr3 = y(ii,9);
    xr = [xr1; xr2; xr3];

    switch(method)
% Controller gains in fuzzy-model-based controller are given as below with different parameter settings;
        case(0)     % parameter settings: sigma_1 = 1e-1; sigma_2 = 1e-1;
F(:,:,1) = [-1.1403e+00 -2.2254e-01 -3.8765e-01; -1.0384e+00 -3.5175e-01 -5.4922e-01; -1.0345e+00 -
1.5969e-01 -6.2389e-01];
```

G(:,:,1) = [-6.8002e-16 -5.9102e-16 1.4869e-15; -7.9019e-16 5.5813e-17 1.2643e-15; -9.3789e-16 -5.5686e-16 1.7721e-15];

F(:,:,2) = [-4.3365e-01 3.4006e-01 -9.1387e-02; -3.5603e-01 2.6167e-01 -1.7156e-01; -3.4103e-01 3.5524e-01 -2.2622e-01];

G(:,:,2) = [4.6627e-15 1.0852e-15 3.1478e-15; 4.2849e-15 1.2336e-15 2.7674e-15; 4.7021e-15 1.3911e-15 2.6294e-15];

F(:,:,3) = [-2.7530e+00 -1.2628e+00 -1.8469e+00; -2.4707e+00 -1.6661e+00 -2.1170e+00; -2.4961e+00 -1.2549e+00 -2.4234e+00];

G(:,:,3) = [1.2974e-15 -7.4556e-16 -1.5843e-15; 1.3266e-15 -4.4159e-16 -1.9478e-15; 1.4296e-15 -7.9754e-16 -1.7477e-15];

F(:,:,4) = [-4.9664e-01 2.2972e-01 5.0946e-01; -5.1533e-01 1.9074e-01 4.8379e-01; -4.9602e-01 2.4277e-01 4.4625e-01];

G(:,:,4) = [-4.8425e-15 2.8775e-15 -4.4699e-15; -4.1778e-15 2.5062e-15 -5.2096e-15; -4.4789e-15 3.2650e-15 -5.2650e-15];

F(:,:,5) = [-2.7687e-01 3.4491e-01 3.1116e-02; -2.9920e-01 2.8003e-01 -1.2672e-03; -2.8089e-01 3.1652e-01 -5.0543e-02];

G(:,:,5) = [-1.3485e-15 1.8531e-15 -3.5293e-15; -1.9470e-15 1.9870e-15 -2.5500e-15; -1.0448e-15 2.4875e-15 -2.0443e-15];

F(:,:,6) = [-9.7936e-01 3.1323e-02 -8.3798e-01; -9.3138e-01 -1.1335e-01 -9.1703e-01; -9.2504e-01 1.5923e-02 -1.0383e+00];

G(:,:,6) = [-1.3964e-15 8.6577e-16 1.7686e-15; -1.6743e-15 6.8643e-16 2.1342e-15; -1.2865e-15 8.3607e-16 1.6819e-15];

```
case(1)
    % sigma_1 = 1e2; sigma_2 = 1e2;
```

F(:,:,1) = [-7.8764e-01 -1.7336e-01 -3.0902e-01; -7.3193e-01 -2.6533e-01 -4.1568e-01; -7.3025e-01 -1.4246e-01 -4.8699e-01];

G(:,:,1) = [-8.6884e-10 1.9783e-09 -1.7434e-09; -1.5510e-09 2.7033e-09 -2.3985e-09; -1.3569e-09 2.8320e-09 -2.5024e-09];

F(:,:,2) = [-4.2613e-01 1.8775e-01 -1.7859e-01; -3.5624e-01 1.1637e-01 -2.4490e-01; -3.4597e-01 2.0161e-01 -2.9304e-01];

```
G(:,:,2) = [-3.5218e-09 -2.1411e-09 -2.8407e-09; -3.8149e-09 -2.1628e-09 -2.9939e-09; -4.2504e-09 -1.9623e-09 -2.0469e-09];
F(:,:,3) = [-1.7733e+00 -7.7868e-01 -1.0933e+00; -1.5753e+00 -1.0420e+00 -1.2604e+00; -1.5934e+00 -7.6613e-01 -1.4639e+00];
G(:,:,3) = [3.0328e-09 -5.0895e-09 5.2575e-09; 3.3430e-09 -5.6021e-09 5.0966e-09; 3.3254e-09 -5.5779e-09 5.7375e-09];
F(:,:,4) = [-4.6896e-01 1.1272e-01 2.7953e-01; -4.8068e-01 6.0424e-02 2.7900e-01; -4.6874e-01 1.0070e-01 2.3111e-01];
G(:,:,4) = [4.7697e-09 -5.6613e-09 2.2104e-09; 4.5124e-09 -6.0336e-09 1.3974e-09; 4.1704e-09 -6.3540e-09 1.2990e-09];
F(:,:,5) = [-2.6943e-01 1.9119e-01 -5.3619e-02; -2.7586e-01 1.3134e-01 -7.4493e-02; -2.5355e-01 1.6345e-01 -1.1754e-01];
G(:,:,5) = [3.5336e-10 1.6530e-09 1.7168e-09; 8.6477e-10 7.5349e-10 2.3677e-09; 2.8370e-11 1.6871e-09 2.0512e-09];
F(:,:,6) = [-7.8604e-01 -8.1921e-02 -7.9382e-01; -7.5554e-01 -1.8954e-01 -8.5116e-01; -7.3845e-01 -8.7065e-02 -9.3149e-01];
G(:,:,6) = [-4.4796e-09 6.8716e-09 -5.9464e-09; -4.4370e-09 7.3136e-09 -6.4558e-09; -4.3508e-09 6.6114e-09 -5.7048e-09];

end

%===== membership Functions =====
w1(1) = 1-1./(1 + exp(-(x1 - 0.045)/0.0045));
w1(2) = 1 - w1(1);

w2(1) = 1-1./(1 + exp(-(x2+0.03)/0.0075));
w2(3) = 1./(1 + exp(-(x2-0.03)/0.0075));
w2(2) = 1 - w2(1) - w2(3);

w3(1) = 1;

np = length(w1)*length(w2)*length(w3);
```

```
    l = 1;
    for i = 1:length(w1)
        for j = 1:length(w2)
            for k = 1:length(w3)
                SW(l) = w1(i)*w2(j)*w3(k);
                l = l + 1;
            end
        end
    end
    FF = 0;
    GG = 0;
    for i = 1:np
        FF = FF + SW(i)*F(:,:,i);
        GG = GG + SW(i)*G(:,:,i);
    end
    u(ii,:) = (FF*(xx - xr) + GG*xr)';
end
```

6) mnp_ode.m

```
% Odinary differential equations
function dx = mnp_ode(t, x, X0, Y0, Z0, X_ref, Y_ref, Z_ref, method)

    x1 = x(1);
    x2 = x(2);
    x3 = x(3);
    xx = [x1; x2; x3];

    l1 = x(4);
    l2 = x(5);
    l3 = x(6);
```

```
    xr1 = x(7);
    xr2 = x(8);
    xr3 = x(9);
    xr = [xr1; xr2; xr3];

    d = 0.005; % the radius of the cross-section of the continuum manipulator; unit: meter.

    switch(method)
% Controller gains in fuzzy-model-based controller are given as below with different parameter settings;
    case(0)     % sigma_1 = 1e-1; sigma_2 = 1e-1;
F(:,:,1) = [-1.1403e+00 -2.2254e-01 -3.8765e-01; -1.0384e+00 -3.5175e-01 -5.4922e-01; -1.0345e+00 -1.5969e-01 -6.2389e-01];
G(:,:,1) = [-6.8002e-16 -5.9102e-16 1.4869e-15; -7.9019e-16 5.5813e-17 1.2643e-15; -9.3789e-16 -5.5686e-16 1.7721e-15];
F(:,:,2) = [-4.3365e-01 3.4006e-01 -9.1387e-02; -3.5603e-01 2.6167e-01 -1.7156e-01; -3.4103e-01 3.5524e-01 -2.2622e-01];
G(:,:,2) = [4.6627e-15 1.0852e-15 3.1478e-15; 4.2849e-15 1.2336e-15 2.7674e-15; 4.7021e-15 1.3911e-15 2.6294e-15];
F(:,:,3) = [-2.7530e+00 -1.2628e+00 -1.8469e+00; -2.4707e+00 -1.6661e+00 -2.1170e+00; -2.4961e+00 -1.2549e+00 -2.4234e+00];
G(:,:,3) = [1.2974e-15 -7.4556e-16 -1.5843e-15; 1.3266e-15 -4.4159e-16 -1.9478e-15; 1.4296e-15 -7.9754e-16 -1.7477e-15];
F(:,:,4) = [-4.9664e-01 2.2972e-01 5.0946e-01; -5.1533e-01 1.9074e-01 4.8379e-01; -4.9602e-01 2.4277e-01 4.4625e-01];
G(:,:,4) = [-4.8425e-15 2.8775e-15 -4.4699e-15; -4.1778e-15 2.5062e-15 -5.2096e-15; -4.4789e-15 3.2650e-15 -5.2650e-15];
F(:,:,5) = [-2.7687e-01 3.4491e-01 3.1116e-02; -2.9920e-01 2.8003e-01 -1.2672e-03; -2.8089e-01 3.1652e-01 -5.0543e-02];
G(:,:,5) = [-1.3485e-15 1.8531e-15 -3.5293e-15; -1.9470e-15 1.9870e-15 -2.5500e-15; -1.0448e-15 2.4875e-15 -2.0443e-15];
F(:,:,6) = [-9.7936e-01 3.1323e-02 -8.3798e-01; -9.3138e-01 -1.1335e-01 -9.1703e-01; -9.2504e-01 1.5923e-02 -1.0383e+00];
```

```
G(:,:,6) = [-1.3964e-15 8.6577e-16 1.7686e-15; -1.6743e-15 6.8643e-16 2.1342e-15; -1.2865e-15 8.3607e-16 1.6819e-15];

    case(1)
        % sigma_1 = 1e2; sigma_2 = 1e2;
F(:,:,1) = [-7.8764e-01 -1.7336e-01 -3.0902e-01; -7.3193e-01 -2.6533e-01 -4.1568e-01; -7.3025e-01 -1.4246e-01 -4.8699e-01];
G(:,:,1) = [-8.6884e-10 1.9783e-09 -1.7434e-09; -1.5510e-09 2.7033e-09 -2.3985e-09; -1.3569e-09 2.8320e-09 -2.5024e-09];
F(:,:,2) = [-4.2613e-01 1.8775e-01 -1.7859e-01; -3.5624e-01 1.1637e-01 -2.4490e-01; -3.4597e-01 2.0161e-01 -2.9304e-01];
G(:,:,2) = [-3.5218e-09 -2.1411e-09 -2.8407e-09; -3.8149e-09 -2.1628e-09 -2.9939e-09; -4.2504e-09 -1.9623e-09 -2.0469e-09];
F(:,:,3) = [-1.7733e+00 -7.7868e-01 -1.0933e+00; -1.5753e+00 -1.0420e+00 -1.2604e+00; -1.5934e+00 -7.6613e-01 -1.4639e+00];
G(:,:,3) = [3.0328e-09 -5.0895e-09 5.2575e-09; 3.3430e-09 -5.6021e-09 5.0966e-09; 3.3254e-09 -5.5779e-09 5.7375e-09];
F(:,:,4) = [-4.6896e-01 1.1272e-01 2.7953e-01; -4.8068e-01 6.0424e-02 2.7900e-01; -4.6874e-01 1.0070e-01 2.3111e-01];
G(:,:,4) = [4.7697e-09 -5.6613e-09 2.2104e-09; 4.5124e-09 -6.0336e-09 1.3974e-09; 4.1704e-09 -6.3540e-09 1.2990e-09];
F(:,:,5) = [-2.6943e-01 1.9119e-01 -5.3619e-02; -2.7586e-01 1.3134e-01 -7.4493e-02; -2.5355e-01 1.6345e-01 -1.1754e-01];
G(:,:,5) = [3.5336e-10 1.6530e-09 1.7168e-09; 8.6477e-10 7.5349e-10 2.3677e-09; 2.8370e-11 1.6871e-09 2.0512e-09];
F(:,:,6) = [-7.8604e-01 -8.1921e-02 -7.9382e-01; -7.5554e-01 -1.8954e-01 -8.5116e-01; -7.3845e-01 -8.7065e-02 -9.3149e-01];
G(:,:,6) = [-4.4796e-09 6.8716e-09 -5.9464e-09; -4.4370e-09 7.3136e-09 -6.4558e-09; -4.3508e-09 6.6114e-09 -5.7048e-09];

end
%===== membership Functions =====
```

```
w1(1) = 1-1./(1 + exp(-(x1 - 0.045)/0.0045));
w1(2) = 1 - w1(1);

w2(1) = 1-1./(1 + exp(-(x2+0.03)/0.0075));
w2(3) = 1./(1 + exp(-(x2-0.03)/0.0075));
w2(2) = 1 - w2(1) - w2(3);

w3(1) = 1;

np = length(w1)*length(w2)*length(w3);

l = 1;
for i = 1:length(w1)
   for j = 1:length(w2)
      for k = 1:length(w3)
         SW(l) = w1(i)*w2(j)*w3(k);
         l = l + 1;
      end
   end
end

FF = 0;
GG = 0;
for i = 1:np
   FF = FF + SW(i)*F(:,:,i);
   GG = GG + SW(i)*G(:,:,i);
End

% control signal generation as shown in Fig. 5-5 and six local linearized model are controlled with the calculated feedback gains. The equation corresponds to Eq. (5-31).
u = FF*(xx - xr) + GG*xr;
```

```
% Elements of the 3×3 Jacobian matrix calculated as below, which depend on the three lengths of three tendons and the cross-section radius of the continuum manipulator.
J11 = Jelement_11(l1, l2, l3, d);
J12 = Jelement_12(l1, l2, l3, d);
J13 = Jelement_13(l1, l2, l3, d);
J21 = Jelement_21(l1, l2, l3, d);
J22 = Jelement_22(l1, l2, l3, d);
J23 = Jelement_23(l1, l2, l3, d);
J31 = Jelement_31(l1, l2, l3, d);
J32 = Jelement_32(l1, l2, l3, d);
J33 = Jelement_33(l1, l2, l3, d);

J = [J11, J12,  J13;
   J21, J22,  J23;
   J31, J32,  J33];

% J = J + [0.2 0 0;  0 0 0;  0 0 0]; We have tested different values to simulate those disturbances, and 0.2 is approximately the mean value to simulate disturbance in our experiment setup.
J = J + 0.2*ones(3,3);

dx(1:3,:) = J*u;
dx(4:6,:) = u(1:3);

% reference model
Ar = zeros(3,3);
Br = 1/60*[X_ref - X0; Y_ref - Y0; Z_ref - Z0];

if t <= 60     % after 60 (seconds), the moving reference target will stop.
   rt = 1;
else
   rt = 0;
end
```

```
dx(7:9,:) = Ar*xr + Br*rt;
```

7) findependent_inv_k.m

```
% to calculate “curvature k” element in the inverse kinematics of the dependent mapping part based on the tip position in Cartesian coordinate frame; the analytical equation is derived with Maple.
function k = findependent_inv_k(x, y, z, d)
t1 = x ^ 2;
t2 = y ^ 2;
t4 = sqrt(t1 + t2);
t5 = z ^ 2;
k = 0.2e1 / (t1 + t2 + t5) * t4;
end
```

8) findependent_inv_l.m

```
% to calculate “length l” element in the inverse kinematics of the dependent mapping part based on the tip position in Cartesian coordinate frame; the analytical equation is derived with Maple.
function l = findependent_inv_l(x, y, z, d)
t1 = x ^ 2;
t2 = y ^ 2;
t3 = z ^ 2;
t4 = t1 + t2 + t3;
t8 = acos(0.1e1 / t4 * (-t1 - t2 + t3));
t11 = sqrt(t1 + t2);
l = 0.1e1 / t11 * t8 * t4 / 0.2e1;
end
```

9) findependent_inv_phi.m

```
% to calculate “bending plane angle phi” element in the inverse kinematics of the dependent mapping part based on the tip position in Cartesian coordinate frame; the analytical equation is derived with Maple.
function phi = findependent_inv_phi(x, y, z, d)
phi = atan(y / x);
```

```
end
```

10) findependent_x.m

```
% to calculate x element in the kinematics of the dependent mapping part based on the configuration of the continuum manipulator; the analytical equation is described in Eq. (4-37).
function x = findependent_x(k, theta, l, d)
t1 = cos(theta);
t3 = cos(k * l);
x = 0.1e1 / k * (0.1e1 - t3) * t1;
end
```

11) findependent_y.m

```
% to calculate y element in the kinematics of the dependent mapping part based on the configuration of the continuum manipulator; the analytical equation is described in Eq. (4-37).
function y = findependent_y(k, theta, l, d)
t1 = sin(theta);
t3 = cos(k * l);
y = 0.1e1 / k * (0.1e1 - t3) * t1;
end
```

12) findependent_z.m

```
% to calculate z element in the kinematics of the dependent mapping part based on the configuration of the continuum manipulator; the analytical equation is described in Eq. (4-37).
function z = findependent_z(k, theta, l, d)
t2 = sin(k * l);
z = 0.1e1 / k * t2;
end
```

13) fspecific_inv_l1.m

```
% to calculate the tendon length of tendon 1 in the inverse kinematics of the specific mapping part based on the configuration of the continuum manipulator; the analytical equation is derived with Maple.
```

```
function a = fspecific_inv_l1(k, phi, l, d)
t1 = sqrt(0.3e1);
t2 = sin(phi);
t3 = t2 * t1;
t4 = d ^ 2;
t5 = k ^ 2;
t7 = cos(phi);
t10 = t7 ^ 2;
t13 = t10 ^ 2;
t17 = sqrt(-0.1e1 / t13 * (0.2e1 * t7 * t3 - 0.2e1 * t10 - 0.1e1) * t5 * t4);
t20 = 0.3e1 * t7;
a = 0.1e1 / (t3 - t20) * (t10 * t17 * t3 - t20 + t3) * l;
end
```

14) fspecific_inv_l2.m

```
% to calculate the tendon length of tendon 2 in the inverse kinematics of the specific mapping part based on the configuration of the continuum manipulator; the analytical equation is derived with Maple.
function b = fspecific_inv_l2(k, phi, l, d)
t1 = sqrt(0.3e1);
t2 = sin(phi);
t3 = t2 * t1;
t4 = d ^ 2;
t5 = k ^ 2;
t7 = cos(phi);
t10 = t7 ^ 2;
t13 = t10 ^ 2;
t17 = sqrt(-0.1e1 / t13 * (0.2e1 * t7 * t3 - 0.2e1 * t10 - 0.1e1) * t5 * t4);
b = -0.1e1 / (t3 - 0.3e1 * t7) * (t10 * t17 * t3 + 0.3e1 * t10 * t7 * t17 - 0.2e1 * t3 + 0.6e1 * t7) * l / 0.2e1;
end
```

15) fspecific_inv_l3.m

```
% to calculate the tendon length of tendon 3 in the inverse kinematics of the specific mapping part based on the configuration of the continuum manipulator; the analytical equation is derived with Maple.
function c = fspecific_inv_l3(k, phi, l, d)
t1 = d ^ 2;
t2 = k ^ 2;
t4 = sqrt(0.3e1);
t5 = sin(phi);
t7 = cos(phi);
t10 = t7 ^ 2;
t13 = t10 ^ 2;
t17 = sqrt(-0.1e1 / t13 * (0.2e1 * t7 * t5 * t4 - 0.2e1 * t10 - 0.1e1) * t2 * t1);
c = -(t10 * t17 - 0.2e1) * l / 0.2e1;
end
```

16) fspecific_k.m

```
% to calculate “curvature k” element in the kinematics of the specific mapping part based on three actuated tendon lengths (a, b, c) and the radius of the cross-section of the continuum manipulator; the analytical equation is described in Eq. (4-39).
function k = fspecific_k(a, b, c, d)
t1 = a ^ 2;
t4 = b ^ 2;
t6 = c ^ 2;
t8 = sqrt(-a * b - a * c - b * c + t1 + t4 + t6);
k = 0.2e1 / (a + b + c) / d * t8;
end
```

17) fspecific_l.m

```
% to calculate “length k” element in the kinematics of the specific mapping part based on three actuated tendon lengths (a, b, c) and the radius of the cross-section of the continuum manipulator; the analytical equation is described in Eq. (4-41).
function l = fspecific_l(a, b, c, d)
l = a / 0.3e1 + b / 0.3e1 + c / 0.3e1;
```

```
end
```

18) fspecific_phi.m

```
% to calculate “bending plane angle phi” element in the kinematics of the specific mapping part based on three actuated tendon lengths (a, b, c) and the radius of the cross-section of the continuum manipulator; the analytical equation is described in Eq. (4-40).
function phi = fspecific_phi(a, b, c, d)
t1 = sqrt(0.3e1);
phi = atan(0.1e1 / (b - c) * (b + c - 2 * a) * t1 / 0.3e1);
end
```

19) Jelement_11.m

```
% to calculate elements of Jacobian matrix based on three actuated tendon lengths (a, b, c) and the radius of the cross-section of the continuum manipulator; the analytical equations are described in Eq. (4-44), Eq. (4-45), and Eq. (4-46).
function t70 = Jelement_11(a, b, c, d)
t1 = a ^ 2;
t4 = b ^ 2;
t6 = c ^ 2;
t7 = -a * b - a * c - b * c + t1 + t4 + t6;
t8 = sqrt(t7);
t9 = a + b + c;
t10 = t9 * t8 / 0.3e1;
t11 = 0.1e1 / d;
t12 = 0.1e1 / t9;
t15 = 0.2e1 * t12 * t11 * t10;
t16 = cos(t15);
t18 = d * (0.1e1 - t16);
t19 = t9 * t18;
t21 = b + c - 2 * a;
t22 = t21 ^ 2;
t25 = 0.1e1 / ((b - c) ^ 2) / 0.9e1;
```

```
t28 = 0.3e1 * t25 * t22 + 0.1e1;

t29 = sqrt(t28);

t33 = 0.1e1 / t8;

t42 = t9 ^ 2;

t51 = sin(t15);

t54 = sqrt(t28);

t55 = 0.1e1 / t54;

t61 = sqrt(t7);

t70 = 0.3e1 * t25 * t33 / t29 / t28 * t21 * t19 + t33 * t55 * t9 * d * t51 * (-t12 * t11 * t33 * t21 * t9 / 0.3e1
- 0.2e1 / t42 * t11 * t10 + 0.2e1 / 0.3e1 * t12 * t11 * t8) / 0.2e1 + 0.1e1 / t61 / t7 * t55 * t21 * t19 / 0.4e1
+ t33 * t55 * t18 / 0.2e1;
end
```

20) Jelement_12.m

```
function t79 = Jelement_12(a, b, c, d)

t1 = a ^ 2;

t4 = b ^ 2;

t6 = c ^ 2;

t7 = -a * b - a * c - b * c + t1 + t4 + t6;

t8 = sqrt(t7);

t9 = a + b + c;

t10 = t9 * t8 / 0.3e1;

t11 = 0.1e1 / d;

t12 = 0.1e1 / t9;

t15 = 0.2e1 * t12 * t11 * t10;

t16 = cos(t15);

t18 = d * (0.1e1 - t16);

t19 = t9 * t18;

t21 = b + c - 2 * a;

t22 = b - c;

t23 = 9 * t22 ^ 2;

t24 = 1 / t23;
```

```
t27 = t21 ^ 2;

t36 = 3 * t24 * t27 + 1;

t37 = sqrt(t36);

t41 = 0.1e1 / t8;

t46 = -a + 2 * b - c;

t51 = t9 ^ 2;

t60 = sin(t15);

t63 = sqrt(t36);

t64 = 0.1e1 / t63;

t70 = sqrt(t7);

t79 = -t41 / t37 / t36 * (6 * t24 * t21 - 6 / t23 / t22 * t27) * t19 / 0.4e1 + t41 * t64 * t9 * d * t60 * (t12 *

t11 * t41 * t46 * t9 / 0.3e1 - 0.2e1 / t51 * t11 * t10 + 0.2e1 / 0.3e1 * t12 * t11 * t8) / 0.2e1 - 0.1e1 / t70 /

t7 * t64 * t46 * t19 / 0.4e1 + t41 * t64 * t18 / 0.2e1;

end
```

21) Jelement_13.m

```
function t79 = Jelement_13(a, b, c, d)

t1 = a ^ 2;

t4 = b ^ 2;

t6 = c ^ 2;

t7 = -a * b - a * c - b * c + t1 + t4 + t6;

t8 = sqrt(t7);

t9 = a + b + c;

t10 = t9 * t8 / 0.3e1;

t11 = 0.1e1 / d;

t12 = 0.1e1 / t9;

t15 = 0.2e1 * t12 * t11 * t10;

t16 = cos(t15);

t18 = d * (0.1e1 - t16);

t19 = t9 * t18;

t21 = b + c - 2 * a;

t22 = b - c;
```

```
t23 = 9 * t22 ^ 2;

t24 = 1 / t23;

t27 = t21 ^ 2;

t36 = 3 * t24 * t27 + 1;

t37 = sqrt(t36);

t41 = 0.1e1 / t8;

t46 = -a - b + 2 * c;

t51 = t9 ^ 2;

t60 = sin(t15);

t63 = sqrt(t36);

t64 = 0.1e1 / t63;

t70 = sqrt(t7);

t79 = -t41 / t37 / t36 * (6 * t24 * t21 + 6 / t23 / t22 * t27) * t19 / 0.4e1 + t41 * t64 * t9 * d * t60 * (t12 *

t11 * t41 * t46 * t9 / 0.3e1 - 0.2e1 / t51 * t11 * t10 + 0.2e1 / 0.3e1 * t12 * t11 * t8) / 0.2e1 - 0.1e1 / t70 /

t7 * t64 * t46 * t19 / 0.4e1 + t41 * t64 * t18 / 0.2e1;

end
```

22) Jelement_21.m

```
function t89 = Jelement_21(a, b, c, d)

t1 = sqrt(0.3e1);

t2 = a ^ 2;

t5 = b ^ 2;

t7 = c ^ 2;

t8 = -a * b - a * c - b * c + t2 + t5 + t7;

t9 = sqrt(t8);

t10 = a + b + c;

t11 = t10 * t9 / 0.3e1;

t12 = 0.1e1 / d;

t13 = 0.1e1 / t10;

t16 = 0.2e1 * t13 * t12 * t11;

t17 = cos(t16);

t18 = 0.1e1 - t17;
```

```
t21 = b - c;

t22 = 0.1e1 / t21 / 0.3e1;

t25 = b + c - 2 * a;

t26 = t25 ^ 2;

t27 = 0.9e1 * t21 ^ 2;

t31 = 0.1e1 + 0.3e1 / t27 * t26;

t32 = sqrt(t31);

t33 = 0.1e1 / t32;

t34 = 0.1e1 / t9;

t35 = t34 * t33;

t39 = d * t18;

t45 = sqrt(t31);

t52 = t25 * t1;

t57 = t10 ^ 2;

t66 = sin(t16);

t70 = t33 * t22;

t77 = sqrt(t8);

t89 = -t35 * t22 * t10 * d * t18 * t1 + t34 / t45 / t31 / t27 / t21 * t10 * t39 * t26 * t1 + t34 * t70 * t10 * d *

t66 * (-t13 * t12 * t34 * t25 * t10 / 0.3e1 - 0.2e1 / t57 * t12 * t11 + 0.2e1 / 0.3e1 * t13 * t12 * t9) * t52 /

0.2e1 + 0.1e1 / t77 / t8 * t70 * t25 * t10 * t39 * t52 / 0.4e1 + t35 * t22 * d * t18 * t52 / 0.2e1;

end
```

23) Jelement_22.m

```
function t100 = Jelement_22(a, b, c, d)

t1 = sqrt(0.3e1);

t2 = a ^ 2;

t5 = b ^ 2;

t7 = c ^ 2;

t8 = -a * b - a * c - b * c + t2 + t5 + t7;

t9 = sqrt(t8);

t10 = a + b + c;

t11 = t10 * t9 / 0.3e1;
```

```
t12 = 0.1e1 / d;

t13 = 0.1e1 / t10;

t16 = 0.2e1 * t13 * t12 * t11;

t17 = cos(t16);

t18 = 0.1e1 - t17;

t21 = b - c;

t22 = 0.1e1 / t21 / 0.3e1;

t25 = b + c - 2 * a;

t26 = t25 ^ 2;

t27 = 0.9e1 * t21 ^ 2;

t28 = 0.1e1 / t27;

t31 = 0.3e1 * t28 * t26 + 0.1e1;

t32 = sqrt(t31);

t33 = 0.1e1 / t32;

t34 = 0.1e1 / t9;

t35 = t34 * t33;

t39 = t25 * t1;

t41 = d * t18 * t39;

t55 = sqrt(t31);

t64 = -a + 2 * b - c;

t69 = t10 ^ 2;

t78 = sin(t16);

t82 = t33 * t22;

t88 = sqrt(t8);

t100 = t35 * t22 * t10 * d * t18 * t1 / 0.2e1 - 0.3e1 / 0.2e1 * t35 * t28 * t10 * t41 - t34 / t55 / t31 * t22 *

(0.6e1 * t28 * t25 - 0.6e1 / t27 / t21 * t26) * t10 * t41 / 0.4e1 + t34 * t82 * t10 * d * t78 * (t13 * t12 * t34

* t64 * t10 / 0.3e1 - 0.2e1 / t69 * t12 * t11 + 0.2e1 / 0.3e1 * t13 * t12 * t9) * t39 / 0.2e1 - 0.1e1 / t88 / t8 *

t82 * t64 * t10 * t41 / 0.4e1 + t35 * t22 * d * t18 * t39 / 0.2e1;

end
```

24) Jelement_23.m

```
function t100 = Jelement_23(a, b, c, d)
```

```
t1 = sqrt(0.3e1);
t2 = a ^ 2;
t5 = b ^ 2;
t7 = c ^ 2;
t8 = -a * b - a * c - b * c + t2 + t5 + t7;
t9 = sqrt(t8);
t10 = a + b + c;
t11 = t10 * t9 / 0.3e1;
t12 = 0.1e1 / d;
t13 = 0.1e1 / t10;
t16 = 0.2e1 * t13 * t12 * t11;
t17 = cos(t16);
t18 = 0.1e1 - t17;
t21 = b - c;
t22 = 0.1e1 / t21 / 0.3e1;
t25 = b + c - 2 * a;
t26 = t25 ^ 2;
t27 = 0.9e1 * t21 ^ 2;
t28 = 0.1e1 / t27;
t31 = 0.3e1 * t28 * t26 + 0.1e1;
t32 = sqrt(t31);
t33 = 0.1e1 / t32;
t34 = 0.1e1 / t9;
t35 = t34 * t33;
t39 = t25 * t1;
t41 = d * t18 * t39;
t55 = sqrt(t31);
t64 = -a - b + 2 * c;
t69 = t10 ^ 2;
t78 = sin(t16);
t82 = t33 * t22;
t88 = sqrt(t8);
```

```
t100 = t35 * t22 * t10 * d * t18 * t1 / 0.2e1 + 0.3e1 / 0.2e1 * t35 * t28 * t10 * t41 - t34 / t55 / t31 * t22 *
(0.6e1 * t28 * t25 + 0.6e1 / t27 / t21 * t26) * t10 * t41 / 0.4e1 + t34 * t82 * t10 * d * t78 * (t13 * t12 * t34
* t64 * t10 / 0.3e1 - 0.2e1 / t69 * t12 * t11 + 0.2e1 / 0.3e1 * t13 * t12 * t9) * t39 / 0.2e1 - 0.1e1 / t88 / t8 *
t82 * t64 * t10 * t41 / 0.4e1 + t35 * t22 * d * t18 * t39 / 0.2e1;
end
```

25) Jelement_31.m

```
function t49 = Jelement_31(a, b, c, d)
t1 = a + b + c;
t3 = -b - c + 2 * a;
t5 = a ^ 2;
t8 = b ^ 2;
t10 = c ^ 2;
t11 = -a * b - a * c - b * c + t10 + t5 + t8;
t12 = sqrt(t11);
t13 = 0.1e1 / t12;
t14 = 0.1e1 / d;
t16 = 0.1e1 / t1;
t19 = t1 * t12 / 0.3e1;
t20 = t1 ^ 2;
t31 = 0.2e1 * t16 * t14 * t19;
t32 = cos(t31);
t38 = sin(t31);
t39 = d * t38;
t41 = sqrt(t11);
t49 = t13 * t1 * d * t32 * (t16 * t14 * t13 * t3 * t1 / 0.3e1 - 0.2e1 / t20 * t14 * t19 + 0.2e1 / 0.3e1 * t16 *
t14 * t12) / 0.2e1 - 0.1e1 / t41 / t11 * t3 * t1 * t39 / 0.4e1 + t13 * t39 / 0.2e1;
end
```

26) Jelement_32.m

```
function t49 = Jelement_32(a, b, c, d)
t1 = a + b + c;
```

```
t3 = -a + 2 * b - c;

t5 = a ^ 2;

t8 = b ^ 2;

t10 = c ^ 2;

t11 = -a * b - a * c - b * c + t10 + t5 + t8;

t12 = sqrt(t11);

t13 = 0.1e1 / t12;

t14 = 0.1e1 / d;

t16 = 0.1e1 / t1;

t19 = t1 * t12 / 0.3e1;

t20 = t1 ^ 2;

t31 = 0.2e1 * t16 * t14 * t19;

t32 = cos(t31);

t38 = sin(t31);

t39 = d * t38;

t41 = sqrt(t11);

t49 = t13 * t1 * d * t32 * (t16 * t14 * t13 * t3 * t1 / 0.3e1 - 0.2e1 / t20 * t14 * t19 + 0.2e1 / 0.3e1 * t16 *

t14 * t12) / 0.2e1 - 0.1e1 / t41 / t11 * t3 * t1 * t39 / 0.4e1 + t13 * t39 / 0.2e1;

end
```

27) Jelement_33.m

```
function t49 = Jelement_33(a, b, c, d)

t1 = a + b + c;

t3 = -a - b + 2 * c;

t5 = a ^ 2;

t8 = b ^ 2;

t10 = c ^ 2;

t11 = -a * b - a * c - b * c + t10 + t5 + t8;

t12 = sqrt(t11);

t13 = 0.1e1 / t12;

t14 = 0.1e1 / d;

t16 = 0.1e1 / t1;
```

```
t19 = t1 * t12 / 0.3e1;

t20 = t1 ^ 2;

t31 = 0.2e1 * t16 * t14 * t19;

t32 = cos(t31);

t38 = sin(t31);

t39 = d * t38;

t41 = sqrt(t11);

t49 = t13 * t1 * d * t32 * (t16 * t14 * t13 * t3 * t1 / 0.3e1 - 0.2e1 / t20 * t14 * t19 + 0.2e1 / 0.3e1 * t16 *

t14 * t12) / 0.2e1 - 0.1e1 / t41 / t11 * t3 * t1 * t39 / 0.4e1 + t13 * t39 / 0.2e1;

end
```

28) fbf_response.m

```
% System responses

clear all

close all

clc


period = 100;

method = 0;


X0 = 0.06;

Y0 = 0.06;

Z0 = 0.03;


X_ref = 0.03;

Y_ref = - 0.06;

Z_ref = 0.135;


d = 0.005; % m


k = findependent_inv_k(X0, Y0, Z0, d);

phi = findependent_inv_phi(X0, Y0, Z0, d);
```

```
l = findependent_inv_l(X0, Y0, Z0, d);

l10 = fspecific_inv_l1(k, phi, l, d);
l20 = fspecific_inv_l2(k, phi, l, d);
l30 = fspecific_inv_l3(k, phi, l, d);

[t1,y1] = ode23(@(t,x)fbf_ode(t, x, X0, Y0, Z0, X_ref, Y_ref, Z_ref, method),[0 period],[X0, Y0, Z0,
l10, l20, l30, X0, Y0, Z0]');

% System states
figure(1);
plot(t1,y1(:,1),'r-',t1,y1(:,2),'b-',t1,y1(:,3),'m-', t1,y1(:,7),'r:',t1,y1(:,8),'b:',t1,y1(:,9),'m:');
ylabel('System states','interpreter','latex');
xlabel('Time $t$ (seconds)','interpreter','latex');
LG = legend('$x(t)$','$y(t)$','$z(t)$','$x_r(t)$','$y_r(t)$','$z_r(t)$');
set(LG,'interpreter','latex');

% Control input
u1 = fbf_cs(t1,y1, X0, Y0, Z0, X_ref, Y_ref, Z_ref, method);
figure(2);
plot(t1,u1(:,1),'r-',t1,u1(:,2),'b-',t1,u1(:,3),'m-');
ylabel('Control input','interpreter','latex');
xlabel('Time $t$ (seconds)','interpreter','latex');
LG = legend('$\dot{\tau}_1(t)$','$\dot{\tau}_2(t)$','$\dot{\tau}_3(t)$');
set(LG,'interpreter','latex');

% 3D
% Workspace
L = 0.1;
figure(3);
switch(method)
    case(0)
```

```
        plot3(y1(:,1),y1(:,2),y1(:,3),'c','LineWidth',2)
end
% plot3(y1(:,1),y1(:,2),y1(:,3),'b','LineWidth',2)

hold on
plot3(y1(:,7), y1(:,8), y1(:,9),'g:','LineWidth',2);
plot3(X0, Y0, Z0, 'Marker','o','MarkerFaceColor','g','MarkerEdgeColor','g','MarkerSize',8);
plot3(y1(end,1),y1(end,2),y1(end,3),
'Marker','o','MarkerFaceColor','r','MarkerEdgeColor','r','MarkerSize',8);
grid on
axis([-0.1*L 0.75*L -0.75*L 0.75*L 0*L 1.5*L])
xlabel('$x(t)$','interpreter','latex');
ylabel('$y(t)$','interpreter','latex');
zlabel('$z(t)$','interpreter','latex');

% to calculate the integral absolute error (IAE) for the numerical comparisons of all the implemented
cotnrollers.
cost = trapz(t1,abs(y1(:,1) - y1(:,7)) + abs(y1(:,2) - y1(:,8)) + abs(y1(:,3) - y1(:,9)))
return

i = 1;
for X0 = - 0.5:0.1:0;
   for Y0 = - 0.5:0.2:0.5;
      for d30 = 0:0.2:1;

          L = 1; % the initial length of the continuum manipulator; unit meter.

          syms y
          alpha_y = solve(L*cos(y + pi/2) == L - (y + pi/2)*sqrt(X0^2 + Y0^2), y);
          alpha0 = double(alpha_y) + pi/2;
          R = L/alpha0;
```

```
            Z0 = R*sin(alpha0) + d30;


            X(i) = X0;
            Y(i) = Y0;
            Z(i) = Z0;
            i = i + 1;
        end
    end
end

figure(3);
scatter3(X,Y,Z,10,'m','fill')

% set(gca,'XLim',[0 1]);
figure(2)
set(gca,'YLim',[-0.1 0.1]);

% l = findobj(gcf, 'type', 'line');
% LG = legend(l([6,3],1), 'Trajectory','Optimized trajectory');
% set(LG,'interpreter','latex','Location','NorthEast');
```

29) ff_response.m

```
% System responses
clear all
% close all
clc

period = 100;
method = 0;
 % the following specify the initial position of the tip of the continuum manipulator; unit: meter.
X0 = 0.06;
Y0 = 0.06;
```

```
Z0 = 0.03;
% the following specify the target position; unit: meter.
X_ref = 0.03;
Y_ref = - 0.06;
Z_ref = 0.135;

d = 0.005; % the radius of the cross-section of the continuum manipulator
 % to calculate the three tendon lengths
k = findependent_inv_k(X0, Y0, Z0, d);
phi = findependent_inv_phi(X0, Y0, Z0, d);
l = findependent_inv_l(X0, Y0, Z0, d);

l10 = fspecific_inv_l1(k, phi, l, d);
l20 = fspecific_inv_l2(k, phi, l, d);
l30 = fspecific_inv_l3(k, phi, l, d);

[t1,y1] = ode23(@(t,x)ff_ode(t, x, X0, Y0, Z0, X_ref, Y_ref, Z_ref, method),[0 period],[X0, Y0, Z0, l10,
l20, l30, X0, Y0, Z0]');
% [t1,y1] = ode23(@(t,x)mnp_ode(t, x, X0, Y0, Z0, X_ref, Y_ref, Z_ref, method),[0 period],[X0, Y0, Z0,
l10, l20, l30, X_ref, Y_ref, Z_ref]');

% System states
figure(1);
plot(t1,y1(:,1),'r-',t1,y1(:,2),'b-',t1,y1(:,3),'m-', t1,y1(:,7),'r:',t1,y1(:,8),'b:',t1,y1(:,9),'m:');
ylabel('System states','interpreter','latex');
xlabel('Time $t$ (seconds)','interpreter','latex');
LG = legend('$x(t)$','$y(t)$','$z(t)$','$x_r(t)$','$y_r(t)$','$z_r(t)$');
set(LG,'interpreter','latex');

% Control input
u1 = ff_cs(t1,y1, X0, Y0, Z0, X_ref, Y_ref, Z_ref, method);
figure(2);
```

```
plot(t1,u1(:,1),'r-',t1,u1(:,2),'b-',t1,u1(:,3),'m-');
ylabel('Control input','interpreter','latex');
xlabel('Time $t$ (seconds)','interpreter','latex');
LG = legend('$\dot{\tau}_1(t)$','$\dot{\tau}_2(t)$','$\dot{\tau}_3(t)$');
set(LG,'interpreter','latex');

% 3D
% Workspace
L = 0.1;
figure(3);
switch(method)
   case(0)
      plot3(y1(:,1),y1(:,2),y1(:,3),'m','LineWidth',2)
end
% plot3(y1(:,1),y1(:,2),y1(:,3),'b','LineWidth',2)

hold on
plot3(y1(:,7), y1(:,8), y1(:,9),'g:','LineWidth',2);
plot3(X0, Y0, Z0, 'Marker','o','MarkerFaceColor','g','MarkerEdgeColor','g','MarkerSize',8);
plot3(y1(end,1),y1(end,2),y1(end,3),
'Marker','o','MarkerFaceColor','r','MarkerEdgeColor','r','MarkerSize',8);
grid on
axis([-0.1*L 0.75*L -0.75*L 0.75*L 0*L 1.5*L])
xlabel('$x(t)$','interpreter','latex');
ylabel('$y(t)$','interpreter','latex');
zlabel('$z(t)$','interpreter','latex');

cost = trapz(t1,abs(y1(:,1) - y1(:,7)) + abs(y1(:,2) - y1(:,8)) + abs(y1(:,3) - y1(:,9)))
return

i = 1;
for X0 = - 0.5:0.1:0;
```

```
    for Y0 = - 0.5:0.2:0.5;
        for d30 = 0:0.2:1;


            L = 1; % m


            syms y
            alpha_y = solve(L*cos(y + pi/2) == L - (y + pi/2)*sqrt(X0^2 + Y0^2), y);
            alpha0 = double(alpha_y) + pi/2;
            R = L/alpha0;


            Z0 = R*sin(alpha0) + d30;


            X(i) = X0;
            Y(i) = Y0;
            Z(i) = Z0;
            i = i + 1;
        end
    end
end


figure(3);
scatter3(X,Y,Z,10,'m','fill')


% set(gca,'XLim',[0 1]);
figure(2)
set(gca,'YLim',[-0.1 0.1]);


% l = findobj(gcf, 'type', 'line');
% LG = legend(l([6,3],1), 'Trajectory','Optimized trajectory');
% set(LG,'interpreter','latex','Location','NorthEast');
```

30) mnp_response.m

```
% System responses
clear all
% close all
clc

period = 100;
method = 0;
 % the following specify the initial position of the tip of the continuum manipulator; unit: meter.
X0 = 0.06;
Y0 = 0.06;
Z0 = 0.03;
 % the following specify the target position; unit: meter.
X_ref = 0.03;
Y_ref = - 0.06;
Z_ref = 0.135;

d = 0.005; % the radius of the cross-section of the continuum manipulator; unit: meter.
 % to calculate the three tendon lengths
k = findependent_inv_k(X0, Y0, Z0, d);
phi = findependent_inv_phi(X0, Y0, Z0, d);
l = findependent_inv_l(X0, Y0, Z0, d);

l10 = fspecific_inv_l1(k, phi, l, d);
l20 = fspecific_inv_l2(k, phi, l, d);
l30 = fspecific_inv_l3(k, phi, l, d);

[t1,y1] = ode23(@(t,x)mnp_ode(t, x, X0, Y0, Z0, X_ref, Y_ref, Z_ref, method),[0 period],[X0, Y0, Z0,
l10, l20, l30, X0, Y0, Z0]');
% [t1,y1] = ode23(@(t,x)mnp_ode(t, x, X0, Y0, Z0, X_ref, Y_ref, Z_ref, method),[0 period],[X0, Y0, Z0,
l10, l20, l30, X_ref, Y_ref, Z_ref]');

% System states
```

```
figure(1);
plot(t1,y1(:,1),'r-',t1,y1(:,2),'b-',t1,y1(:,3),'m-', t1,y1(:,7),'r:',t1,y1(:,8),'b:',t1,y1(:,9),'m:');
ylabel('System states','interpreter','latex');
xlabel('Time $t$ (seconds)','interpreter','latex');
LG = legend('$x(t)$','$y(t)$','$z(t)$','$x_r(t)$','$y_r(t)$','$z_r(t)$');
set(LG,'interpreter','latex');

% Control input
u1 = mnp_cs(t1,y1,method);
figure(2);
plot(t1,u1(:,1),'r-',t1,u1(:,2),'b-',t1,u1(:,3),'m-');
ylabel('Control input','interpreter','latex');
xlabel('Time $t$ (seconds)','interpreter','latex');
LG = legend('$\dot{\tau}_1(t)$','$\dot{\tau}_2(t)$','$\dot{\tau}_3(t)$');
set(LG,'interpreter','latex');

% 3D
% Workspace
L = 0.1;
figure(3);
switch(method)
   case(0)
      plot3(y1(:,1),y1(:,2),y1(:,3),'b','LineWidth',2)
   case(1)
      plot3(y1(:,1),y1(:,2),y1(:,3),'k','LineWidth',2)
   case(2)
      plot3(y1(:,1),y1(:,2),y1(:,3),'k','LineWidth',2)
   case(3)
      plot3(y1(:,1),y1(:,2),y1(:,3),'k','LineWidth',2)

end
% plot3(y1(:,1),y1(:,2),y1(:,3),'b','LineWidth',2)
```

```
hold on
plot3(y1(:,7), y1(:,8), y1(:,9),'g:','LineWidth',2);
plot3(X0, Y0, Z0, 'Marker','o','MarkerFaceColor','g','MarkerEdgeColor','g','MarkerSize',8);
plot3(y1(end,1),y1(end,2),y1(end,3),
'Marker','o','MarkerFaceColor','r','MarkerEdgeColor','r','MarkerSize',8);
grid on
axis([-0.1*L 0.75*L -0.75*L 0.75*L 0*L 1.5*L])
xlabel('$x(t)$','interpreter','latex');
ylabel('$y(t)$','interpreter','latex');
zlabel('$z(t)$','interpreter','latex');

if method == 0

    sol = ode23(@(t,x)mnp_ode(t, x, X0, Y0, Z0, X_ref, Y_ref, Z_ref, method),[0 period],[X0, Y0, Z0,
l10, l20, l30, X0, Y0, Z0]');
    % [t1,y1] = ode23(@(t,x)mnp_ode(t, x, X0, Y0, Z0, X_ref, Y_ref, Z_ref, method),[0 period],[X0, Y0,
Z0, l10, l20, l30, X_ref, Y_ref, Z_ref]');

    t_number = 11;
    m_number = 50;

    t_via = linspace(0, 65, t_number);
    X_via = deval(sol,t_via,1);
    Y_via = deval(sol,t_via,2);
    Z_via = deval(sol,t_via,3);

    for j = 1:t_number

        Xt = X_via(j);
        Yt = Y_via(j);
```

```
        Zt = Z_via(j);

        k = findependent_inv_k(Xt, Yt, Zt, d);
        phi = findependent_inv_phi(Xt, Yt, Zt, d);
        l = findependent_inv_l(Xt, Yt, Zt, d);

        l_via = linspace(0, l, m_number);

        for jj = 1:m_number
            lm = l_via(jj);

            xm(jj) = findependent_x(k, phi, lm, d);
            ym(jj) = findependent_y(k, phi, lm, d);
            zm(jj) = findependent_z(k, phi, lm, d);

        end

        figure(3);
        plot3(xm,ym,zm,'m','LineWidth',2)

    end

end

cost = trapz(t1,abs(y1(:,1) - y1(:,7)) + abs(y1(:,2) - y1(:,8)) + abs(y1(:,3) - y1(:,9)))

return

i = 1;
for X0 = - 0.5:0.1:0;
    for Y0 = - 0.5:0.2:0.5;
        for d30 = 0:0.2:1;
```

```
            L = 1; % m

            syms y
            alpha_y = solve(L*cos(y + pi/2) == L - (y + pi/2)*sqrt(X0^2 + Y0^2), y);
            alpha0 = double(alpha_y) + pi/2;
            R = L/alpha0;

            Z0 = R*sin(alpha0) + d30;

            X(i) = X0;
            Y(i) = Y0;
            Z(i) = Z0;
            i = i + 1;
        end
    end
end

figure(3);
scatter3(X,Y,Z,10,'m','fill')

% set(gca,'XLim',[0 1]);
figure(2)
set(gca,'YLim',[-0.1 0.1]);

% l = findobj(gcf, 'type', 'line');
% LG = legend(l([6,3],1), 'Trajectory','Optimized trajectory');
% set(LG,'interpreter','latex','Location','NorthEast');
```

31) membership.m

```
clear all
close all
```

```
X = 0.015:0.001:0.075;
w1 = 1-1./(1 + exp(-(X - 0.045)/0.0045));
w2 = 1 - w1;

subplot(1,2,1);
plot(X,w1,'r',X,w2,'b')
ylabel('Membership functions','interpreter','latex');
xlabel('$x$','interpreter','latex');
LG = legend('around 0.015','around 0.075');
set(LG,'interpreter','latex');

%% set Front
 set(gca,'FontSize',20)
h = legend('show');
set(h,'FontSize',20)
xl = get(gca,'xlabel');
set(xl,'FontSize',22)
yl = get(gca,'ylabel');
set(yl,'FontSize',22)

Y = -0.075:0.001:0.075;
w1 = 1-1./(1 + exp(-(Y+0.03)/0.0075));
w3 = 1./(1 + exp(-(Y-0.03)/0.0075));
w2 = 1 - w1 - w3;

subplot(1,2,2);
plot(Y,w1,'r',Y,w2,'g',Y,w3,'b')
ylabel('Membership functions','interpreter','latex');
xlabel('$y$','interpreter','latex');
LG = legend('around $-$0.075','around 0','around 0.075');
set(LG,'interpreter','latex');
```

```
%% set Front
set(gca,'FontSize',20)
h = legend('show');
set(h,'FontSize',20)
xl = get(gca,'xlabel');
set(xl,'FontSize',22)
yl = get(gca,'ylabel');
set(yl,'FontSize',22)
```

32) mnp_response.m

mnp_lmi_pdc.m

```
% PDC, SOS-based stability conditions for polynomial fuzzy-model-based control system
clear all
% close all

method = 2;

% Symbolic variables
X1 = sym('X1','real');
X2 = sym('X2','real');
X3 = sym('X3','real');

varX = [X1; X2; X3];

% Example Manipulator
f1 = [0.015  0.075]; % X
% f1 = [0.3]; % X

f2 = [-0.075 0 0.075]; % Y
% f2 = [-0.5  0.5]; % Y
% f2 = [0]; % Y
```

```
% f3 = [0.015  0.15];
f3 = [0.075];

d = 0.005; % m

ii = 1;
for i1 = 1:length(f1)
    for i2 = 1:length(f2)
        for i3 = 1:length(f3)

            x = f1(i1);
            y = f2(i2);
            z = f3(i3);

            k = findependent_inv_k(x, y, z, d);
            phi = findependent_inv_phi(x, y, z, d);
            l = findependent_inv_l(x, y, z, d);

            l1 = fspecific_inv_l1(k, phi, l, d);
            l2 = fspecific_inv_l2(k, phi, l, d);
            l3 = fspecific_inv_l3(k, phi, l, d);

            J11 = Jelement_11(l1, l2, l3, d);
            J12 = Jelement_12(l1, l2, l3, d);
            J13 = Jelement_13(l1, l2, l3, d);
            J21 = Jelement_21(l1, l2, l3, d);
            J22 = Jelement_22(l1, l2, l3, d);
            J23 = Jelement_23(l1, l2, l3, d);
            J31 = Jelement_31(l1, l2, l3, d);
            J32 = Jelement_32(l1, l2, l3, d);
            J33 = Jelement_33(l1, l2, l3, d);
```

```
            A(:,:,ii) = zeros(3,3);
            B(:,:,ii) = [J11, J12,  J13;
                J21, J22,  J23;
                J31, J32,  J33];

            ii = ii + 1;
        end
    end
end

Ar = zeros(3,3);
  % the following specify the initial position of the tip of the continuum manipulator; unit: meter.
X0 = 0.06;
Y0 = 0.06;
Z0 = 0.03;
% the following specify the target position; unit: meter.
X_ref = 0.03;
Y_ref = - 0.06;
Z_ref = 0.135;

Br = 1/60*[X_ref - X0; Y_ref - Y0; Z_ref - Z0];

% Dimensions
[trash1, dimen_N, np] = size(A);
[trash2, dimen_m, trash3] = size(B);

nc = np; % number of rules for controllers

% ==========================================
% Initialize the sum of squares program
```

```
prog = sosprogram([varX]);

% Matrix variables
VECX = monomials(varX,0); % X
[prog,XP1] = sospolymatrixvar(prog,[VECX],[dimen_N dimen_N],'symmetric');

VECN = monomials(varX,0:1:0); % N
for i1 = 1:nc
   [prog,M(:,:,i1)] = sospolymatrixvar(prog,[VECN],[dimen_m dimen_N]);
   [prog,N(:,:,i1)] = sospolymatrixvar(prog,[VECN],[dimen_m dimen_N]);
end

% ==========================================
% Predefined scalar polynomials for positive definite rather than semi-positive
varsigma1 = 1e-3;
varsigma2 = 1e-3;
varsigma3 = 1e-3;

% ==========================================
%SOS conditions
expr = XP1 - varsigma1*eye(dimen_N);
prog = sosmatrixineq(prog,expr,'Mineq');

switch(method)

   case(1)
      % K. Tanaka, 1998, Fuzzy Regulators and Fuzzy Observers: Relaxed Stability Conditions and LMI-Based Designs
      for i1 = 1:np
         for i2 = 1:np
            if i1 <= i2
               Qij = A(:,:,i1)*XP1 + B(:,:,i1)*N(:,:,i2) + transpose(A(:,:,i1)*XP1 + B(:,:,i1)*N(:,:,i2));
```

```
            Qji = A(:,:,i2)*XP1 + B(:,:,i2)*N(:,:,i1) + transpose(A(:,:,i2)*XP1 + B(:,:,i2)*N(:,:,i1));
            expr = - (Qij + Qji);

            expr = expr - varsigma3*eye(dimen_N);
            prog = sosmatrixineq(prog,expr,'Mineq');
        end
    end
end

case(2)

    sigma_1 = 1e-1;
    sigma_2 = 1e-1;

for i1 = 1:np
    for i2 = 1:np
        if i1 <= i2

            k1 = i1; k2 = i2;
            phi_1 = A(:,:,k1)*XP1 + B(:,:,k1)*M(:,:,k2);
            phi_2 = (A(:,:,k1) - Ar)*XP1 + B(:,:,k1)*N(:,:,k2);
            phi_3 = - Br;
            Xi_ij = [phi_1 + transpose(phi_1) + eye(dimen_N),  phi_2,  phi_3;
                transpose(phi_2),  -sigma_1^2*eye(dimen_N),  zeros(3,1);
                transpose(phi_3),  zeros(1,3),  -sigma_2^2];

            k1 = i2; k2 = i1;
            phi_1 = A(:,:,k1)*XP1 + B(:,:,k1)*M(:,:,k2);
            phi_2 = (A(:,:,k1) - Ar)*XP1 + B(:,:,k1)*N(:,:,k2);
            phi_3 = - Br;
            Xi_ji = [phi_1 + transpose(phi_1) + eye(dimen_N),  phi_2,  phi_3;
                transpose(phi_2),  -sigma_1^2*eye(dimen_N),  zeros(3,1);
```

```
                transpose(phi_3),  zeros(1,3),  -sigma_2^2];


            expr = - (Xi_ij + Xi_ji);


            expr = expr - varsigma3*eye(2*dimen_N + 1);
            prog = sosmatrixineq(prog,expr,'Mineq');
        end
      end
    end
end

% ==========================================
% Call solver
[prog, info] = sossolve(prog);
 % Solutions
SOLXP1 = sosgetsol(prog,XP1);

if det(SOLXP1)~=0
   for i1 = 1:nc
      F(:,:,i1) = sosgetsol(prog,M(:,:,i1));
      F(:,:,i1) = F(:,:,i1)/SOLXP1;
      fprintf('F(:,:,%d) = [%.4e %.4e %.4e; %.4e %.4e %.4e; %.4e %.4e %.4e];\n', i1, double(F(:,:,i1))')

      G(:,:,i1) = sosgetsol(prog,N(:,:,i1));
      G(:,:,i1) = G(:,:,i1)/SOLXP1;
      fprintf('G(:,:,%d) = [%.4e %.4e %.4e; %.4e %.4e %.4e; %.4e %.4e %.4e];\n', i1, double(G(:,:,i1))')
   end
end
```

33) B.mat

```
% 3x3x6 double
val(:,:,1) =
```

-0.5013 5.7726 -5.1456

-3.9060 2.1090 1.1683

-6.1484 2.2873 4.4897

val(:,:,2) =

0.0649 4.3379 -4.2080

-5.0660 2.5330 2.5330

0.3247 -0.8269 1.4764

val(:,:,3) =

-0.5013 -5.1456 5.7726

3.9060 -1.1683 -2.1090

-6.1484 4.4897 2.2873

val(:,:,4) =

-2.2373 5.6955 -2.9647

-5.1645 4.4155 0.2555

-6.0364 -2.1052 8.6351

val(:,:,5) =

0.2122 3.3592 -2.9348

-6.3662 3.1831 3.1831

0.2122 -5.3011 5.7255

val(:,:,6) =

2.5663 3.2937 -5.3665

-4.8355 0.5845 4.7445

6.3654 -8.3061 2.4342

34) Br.mat

```
% 3x1 double; to assign the value of Br matrix;
```

-0.000500000000000000

-0.002000000000000000

0.001750000000000000

Appendix E. ROS Interconnections

During the experiments of the multi-tendon driven continuum manipulator tracking tasks, all the implemented controllers and reference trajectory generator are running in Robot Operating System (ROS) environment. Figure E-1 illustrates the interconnections of the ROS setup for controlling the three DC motors with the Aurora® tracking information.

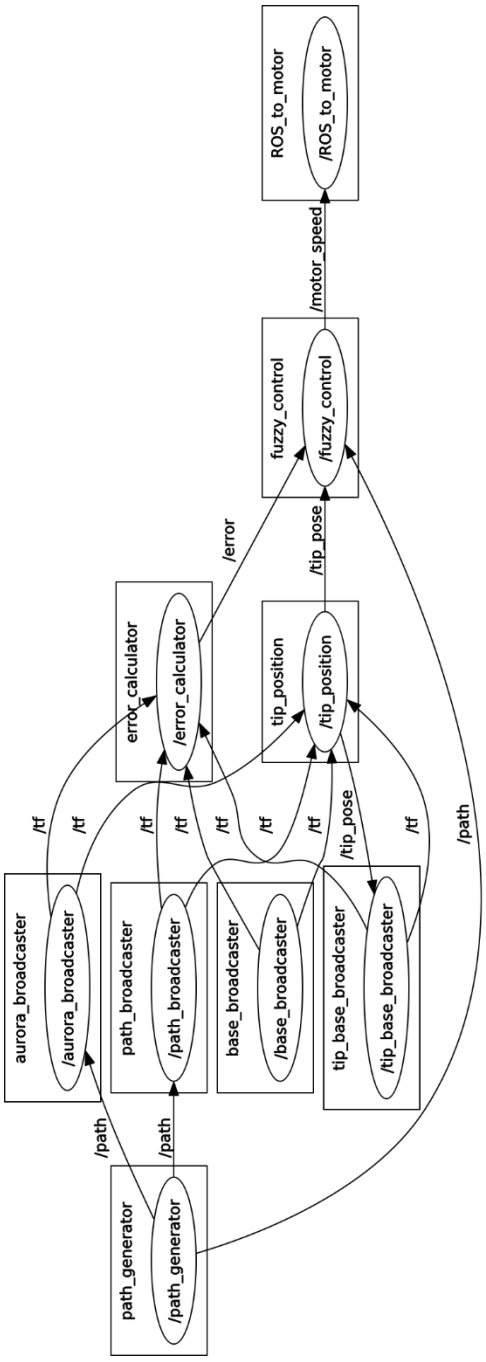


Figure E-1 ROS graph of the experimental setup.